CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Stochastic Programming Approach for Coordinated Contract Decisions in a Make-to-Order Manufacturing Supply Chain

Yan Feng
Alain Martel
Sophie D'Amours
Robert Beauregard

March 2010

CIRRELT-2010-11

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Stochastic Programming Approach for Coordinated Contract Decisions in a Make-to-Order Manufacturing Supply Chain

Yan Feng[1,2,*], Alain Martel[1,3], Sophie D'Amours[1,2], Robert Beauregard[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Département de génie mécanique, Université Laval, Pavillon Adrien-Pouliot, 1065, avenue de la médecine, Québec, Canada G1V 0A6

[3] Département des opérations et systèmes de décision, Pavillon Palasis-Prince, 2325, rue de la Terrasse, Université Laval, Québec, Canada G1V 0A6

[4] Département des sciences du bois et de la forêt, Pavillon Abitibi-Price, 2405, rue de la Terrasse, Université Laval, Québec, G1V 0A6

Abstract. This article proposes a stochastic programming approach for coordinated contract design, allocation and selection decisions, from a manufacturer's point of view, in a three-tier manufacturing supply chain. In a capacitated make-to-order manufacturing system, the manufacturer wishes to offer different customer-contracts to satisfy their needs, to accept the contracts that optimize resource capacity allocations, and to select supplier-contracts that guarantee the satisfaction of the demand in order to maximize profits. Using a two-stage stochastic programming model with recourse, these decisions are addressed under a stochastic economic, market, supply, and system environment. The computational results show that the proposed model provides more realistic and robust solutions, with expected 12% performance improvement over the solutions provided by a deterministic mixed integer programming model.

Keywords. Supply chain coordination, contract design, sales and operations planning (S&OP), make-to-order manufacturing, stochastic programming, oriented strand board (OSB) industry.

Acknowledgements. This project would not have been possible without the collaboration of FOR@C's partners, and the financial support of Natural Sciences and Engineering Council of Canada (NSERC grant # CAP 248987-01).

---

* Corresponding author: Yan.Feng@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec,
Bibliothèque et Archives Canada, 2010

1. Introduction

Effective supply chain management requires collaboration and coordination between independently managed business entities along the supply chain. This function is generally governed by supply chain contracts (or agreements). There is a growing body of research on supply chain contracts defining relationships between supply chain partners. Most of the existing literature focuses on two-tier supplier-buyer contracts, with few exceptions that expend the contract decisions to a general supply chain network context (D'Amours et al. 2000). We consider a three-tier manufacturing supply chain, as illustrated in Figure 1, where a multi-site manufacturer purchases various raw materials from multiple suppliers, and produces different specialty and commodity products for a random demand and price market. Thus, there are contract relationships at both demand and supply ends of the supply chain.

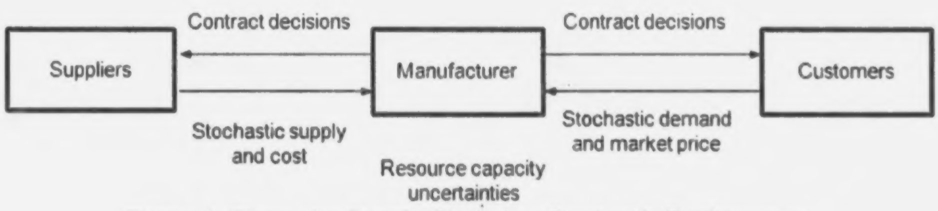

Figure 1. Three-tier Supply Chain in a Stochastic Environment

Generally, contract decisions are made at the beginning of the planning horizon. In a capacitated make-to-order manufacturing system, this decision involves selecting the contract customers so that their demand satisfaction is guaranteed and selecting the contract suppliers so that the raw material supplies are guaranteed while the manufacturer's financial objectives are reached. The manufacturer signs a contract with a customer only if there is enough capacity to satisfy the customer's demand. Hence, the manufacturer would typically allocate a certain proportion of the capacity to contract customers, keeping a capacity buffer for unexpected demand increases and/or to serve spot markets possibly for greater profitability. From a financial point of view, if the market becomes stronger, preserving or increasing contract sales would possibly cause contract demand backlogs and limit the opportunities for greater profitability. However, if the market weakens, reducing contract sales would potentially put the manufacturer at risk of incurring lower profits. Similar scenarios apply to the supply end where the manufacturer has the options to purchase raw materials through contract suppliers or from the open market (spot market) where greater discounts may be possible.

Once the demand contracts are signed with the customers, both contract and capacity allocations are determined, which blocks a proportion of the capacity for the entire contract duration term.

Consequently, sub-optimal contract decisions would have significant impacts on both contract and spot sales, production and logistic performances, as well as raw material supply. Therefore, the demand contract decisions cannot be made in isolation. They must be coordinated both horizontally across different functions of the supply chain and vertically anticipating the impacts on the down stream operational decisions. This is a typical hierarchical planning problem where one has a decision time hierarchy. The objective of this article is to develop an optimization model to help the manufacturer to coordinate contract decisions at both demand and supply ends, and to allocate capacity, in such a way as to maximize the manufacturer's profitability while hedging against uncertainty.

In reality, during the course of the contract period, many uncertain events may happen related to economic conditions, market prices, customer demand, supply availability, and system capacity due to machine failures. This renders the decision-makers under significant risks when making contract decisions. In order to making robust contract decisions that are capable of coping with various uncertainties, a mathematical model that can anticipate the system performances under different plausible futures is required. In this article, we propose a stochastic programming approach to address coordinated contract design, allocation and selection decisions in a three-tier manufacturing supply chain. The research was carried out based on a real case in the Oriented Strand Board (OSB) industry.

OSB is a wood based structural panel widely used in North America as building material for wall, roof, and floor sheathings as well as I-joists. It is made of wood strands mixed with synthetic resins and wax compressed under high temperature and pressure in a hot press. The production is carried out on a highly automated production line, either in batch or in a continuous manner, depending on the type of hot press used. The production line is capable of making a wide range of OSB products including specialty and commodity products with different physical and mechanical properties. The products are mainly sold on contract and non-contract basis, in different markets, to four categories of customers: manufacturers (producing houses or house components), distributors, wholesalers, and retailers. The demand is highly seasonal with strong correlations with the activities in the building construction industry, whereas the supply, particularly for wood logs from forests, is affected by seasonal harvesting operations and long replenishment lead-times.

We address the three-tier supply chain contract design, allocation, and selection problem from the manufacturer's point of view. The manufacturer wishes to offer different contracts to suit the customers' needs and effectively allocate its resource capacities to the right customers, products, and locations. Among different types of contracts found in the literature and practice, we consider

four types of contracts that the manufacturer may offer: i) price-only, ii) periodical minimum quantity commitment, iii) periodical commitment with order band, iv) periodical stationary commitment. The manufacturer also needs to determine which supply contract to accept from which suppliers in order to guarantee the satisfaction of the contract and non-contract demand at lowest procurement cost. In this study, we limit the supply contracts to total minimum quantity commitments with different terms and prices from different suppliers.

We begin the article with a literature review in Section 2 to establish the foundation for this research. In Section 3, the problem definitions are provided with supply chain characteristics, economic, market, and contract descriptions. The two-stage stochastic programming model is presented in Section 4, followed by the solution approach in Section 5. Scenario sampling and model implementation are discussed in Section 6 with computational results being presented in Section 7. Section 8 provides the concluding remarks and future research directions.

2. Literature Review

Since the 1990s, extensive work has been carried out in the general area of supply chain contracts. Tsay et al. (1999) and Cachon (2003) presented detailed reviews of various forms of contracts. Among them, the price-only contract is probably one of the simplest dominant forms of contracts used in practice. In this type of contract, a manufacturer quotes a unit wholesale price to a customer, and the customer has the flexibility to order any quantity in each period during the contract duration term. Lariviere (1999) pointed out that in price only contracts, suppliers tend to sell at a wholesale price above the production marginal cost, which induces the retailer to set a retail price above what an integrated firm would charge (also known as double marginalization), which could result in lower sales and profits than what an integrated channel would achieve. Lariviere and Porteus (2001) studied the price-only contract in a two-echelon distribution channel with a supplier selling to a single retailer facing a single-period newsvendor problem. It was concluded that price-only contracts cannot provide supply chain coordination.

Another widely applied form of contract is quantity discount contract. This type of contract introduces price incentives so as to stimulate sales and maximize supplier's profits. Monahan (1984) studied a single period quantity discount contract between a buyer and a supplier assuming the buyer is likely to react to any supplier's discount proposal. Weng (1995) investigated the effects of a single period quantity discount model on channel coordination and profit maximization. The analysis shows that quantity discount contracts do not guarantee joint profit maximization. However, channel coordination can be reached by employing quantity discounts and franchise fees simultaneously. Munson and Rosenblatt (2001) studied a quantity discount model in a three-echelon supply chain with the middle echelon being the decision maker

offering different discount schemes. Clearly, discounts can be offered in combination with different contract forms where price incentives are necessary.

Under total minimum quantity commitment contracts, while a supplier offers a discounted price, a total minimum quantity commitment is required and, as the total minimum commitment increases, the unit price decreases. The buyer commits to purchase, during the entire contract horizon, at least the minimum quantity at the discounted price. There is no restriction on the maximum amount that can be purchased, nor requirement on the exact amount purchased in each period. Observations found that in a stochastic demand environment, the buyer inclines to purchase exactly its demand requirement, thus passing its demand uncertainties onto the supplier. Nevertheless, total minimum commitment contracts have been widely used as suppliers wish to increase market shares by locking-in buyers to commit to purchase in a longer term. On the other hand, if there is any uncertainty in the supply process, a buyer may wish to enter into such a contract to ensure long term supply (Anupindi and Bassok 1999). Bassok and Anupindi (1997) provided early work on supply contracts with total minimum quantity commitment for a single-product periodical review inventory problem with random demand. By studying a multi-period setting, Anupindi and Bassok (1999) argue that although the total minimum quantity commitment provides buying flexibility at discounted price, it may lead to supplier loss.

One of the remedy to this problem is the periodical commitment contract. Unlike the total minimum commitment contract, the periodical commitment contract imposes restrictions on periodical purchases and, thus, reduces the uncertainty in the order process. This contract may take various forms depending on the nature of periodical commitments and the flexibility offered. Broadly, the commitments could be stationary or dynamic. Stationary commitment contracts were analysed by Moinzadeh and Nahmias (2000) and Anupindi and Akella (1997). With a stationary commitment, a buyer is required to purchase a fixed minimum amount in each period. Discounts are given based on the level of minimum commitment. Additional units can be purchased but at an extra cost and the delivery may be delayed. This contract provides a greater level of demand certainty for the supplier and just-in-time delivery for the customer. With dynamic commitments, the minimum amount can be updated periodically in a rolling horizon manner. The use of rolling horizon procedures in contract based planning was investigated by D'Amours et al. (2000) in a manufacturing supply chain context. More recently, Lian and Deshmukh (2009) studied a rolling horizon planning contract with dynamic commitment and quantity flexibility between a buyer and a supplier for a single product. The flexibility in the contract can be offered in the form of an order band, where all order quantities are required to be within stationary lower and upper limits. Order-band contracts were initially studied by Kumar (1992) and Anupindi (1993) in a game-theoretic setting. Scheller-Wolf and Tayur (1998)

extended the study in a Markovian demand environment. These contracts can also offer quantity flexibility through changing minimum and maximum limits revised in percentages that vary in accordance with the number of periods away from the delivery (Anupindi and Bassok 1999). Earlier studies on quantity flexibility contracts were published by Bassok and Anupindi (1997), Tsay (1999), and Tsay and Lovejoy (1999).

In supply chain contract design, a decision-maker has to determine what types of contract to offer, with what terms and conditions, and what reactions are possible from the customers. To tackle these questions, most of the researchers adopted an agent-based approach focusing on a contract between a buyer and a supplier. The buyer's optimization problem is solved first to determine his optimal order quantity according to the contract offered by the supplier. Then the supplier's optimization problem is solved for the buyer's optimal order quantity to determine the optimal supply contract. A Nash-equilibrium is reached and the costs (profits) of the buyer and supplier are examined to determine the optimal contract setting (Corbett and Tang 1999, Schneeweiss et al. 2004). When a manufacturer serves several customer-product-locations competing for its limited capacity, such as in our case, contract decisions becomes more complex. Unfortunately, such concerns have not yet been considered in most of the literature. One of the difficulties of addressing the coordinated contract design and allocation problem in a single supplier serving multiple customers is the ability to understand the possible reactions of the customers to the contract(s) offered. Consider that, instead of addressing the supplier's contract design problem based on a single-factor customer cost structure, like what has been assumed in most of the contract analysis and design problems, it is possible that the customer's choice of a contract is affected by several factors, the combined attributes of the contract policy, for instance. In this context, whether or not the customer will choose an offered contract policy is a probabilistic discrete choice problem, depending on the economic evaluation of the customer, as well as his perceived product qualities, the services provided, and socio-economic considerations. According Ben-Akiva and Lerman (1994) and Vila et al. (2007), such probability may be determined based on random utility theory using a logit discrete choice model. Vila et al. (2007) applied this method to determine the customer-contract choice probabilities for several customers, where the customers' reactions to the contracts offered are anticipated in a strategic supply chain design model. Similar approaches are adopted in bidding problems for a manufacturer facing multiple customer classes, as shown in Easton and Moodie (1999) and Watanapa and Techanitisawad (2004).

Furthermore, in the contract analysis and design problems, most of the models proposed assume a deterministic structure, with a few exceptions found in van Delft and Vial (2001), Zou et al. (2008), and Xu and Nozick (2008). Van Delft and Vial (2001) presented a stochastic

programming approach for multi-period supply contract analysis between a buyer and a supplier. Zou et al. (2008) proposed a stochastic dynamic programming approach to design a supply contract between an assembler and two suppliers in an assembly system. Xu and Nozick (2008) proposed a two-stage stochastic model for facility location and network design with the possibility of using option contracts to hedge against uncertain events which could cause capacity loss at one or several suppliers in a geographic area.

In this study, the contract design and allocation problem at the demand end is addressed, and the possible customer reactions to a contract offer are anticipated through probabilistic customer-contract choice analysis. The stochastic programming model presented in the following section is based on the deterministic model for multi-site supply chain sales and operations planning (SC-S&OP) developed by Feng et al. (2008) and the case therein.

3. Problem Definition

3.1 Supply Chain Characteristics

In this study, we consider a manufacturing supply chain network, consisting of a manufacturer, and several customers, suppliers and third-party distribution centres (DCs), as shown in Figure 2. The manufacturer has many production sites scattered in different regions. We define $\mathcal{V}=(S,M,D,C)$ as the set of network nodes (vertices), where S, M, D, and C are subsets associated to raw material suppliers, manufacturing sites, DCs, and customers respectively. Let $\mathcal{R}=(S\times M,M\times D,M\times C,D\times C)$ be the set of inbound and outbound arcs, corresponding to ordered pairs of elements of $\mathcal{V}$. The manufacturer produces both specialty and commodity products. The specialty products ($i\in I_{spe}$) are sold through contract agreements, and commodity products ($i\in I_{com}$) can be sold through contract agreements or on the spot market. Both contract and spot market demands are highly seasonal. Customers ordering specialty products prefer a contract relationship in order to secure their supply. If the contract is not awarded, the customer is likely to seek other sources from competitors. A customer ordering commodity products may also choose a contract relationship, however she may purchase from the manufacturer through spot sales when a contract is not signed. The spot market is considered as a recourse which can absorb any production amount.

Each manufacturing site $m\in M$ has a single capacitated production line producing a set $I=I_{spe}\cup I_{com}$ of product families[1] on an MTO basis with small on-site inventory capacity. Every manufacturing site can produce all products $i\in I$ and everyone can contribute to satisfy a given

[1] In the reminder of the text, the word "product" should be interpreted as "product family".

contract subject to capacity constraints. However a contract may be satisfied more economically by one site than others due to its efficiency and location. We assume that production capacity is affected by unexpected machine down time, and hence, for plant m in planning period $t \in T$, it is an independent random variable K_{mt} with cumulative distribution function $F_{K_{mt}}(.)$.

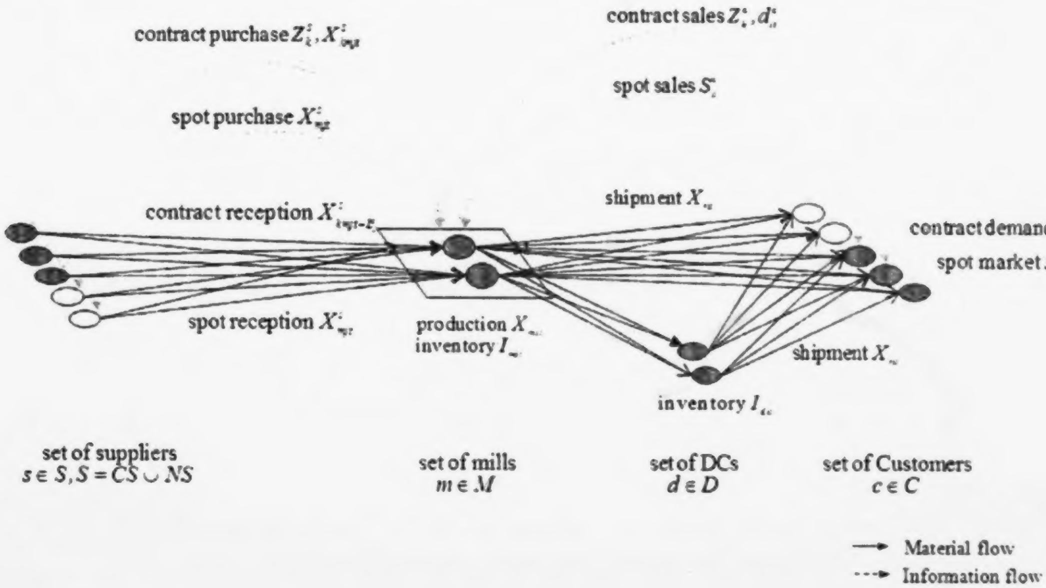

Figure 2. Contract Relationships in a Three-tier Manufacturing Supply Chain Network

The production of each product $i \in I$ consumes a set of J raw materials with different ratios defined by a product recipe. The manufacturer purchases these raw materials from suppliers $s \in S$, including several potential contract suppliers ($CS \subset S$) as well as non-contract spot market suppliers ($NS \subset S$). Suppliers have different procurement lead-times L^s_j for raw material $j \in J$. Raw materials are classified into different categories and stored using different storage technologies. Let G be the set of storage technologies and g, a particular technology with storage capacity KI_{mg} for mill m. Also, let $J_g \subset J$ be the subset of raw materials that can be stored with technology g. The raw material inventory is managed internally complying with safety stock policies. We assume that inbound raw material shipments are carried out by the suppliers, and that their shipping costs are included in the procurement costs.

The outbound shipments of the products from the manufacturing sites to the customers are carried out by third party logistic (3PL) providers, either directly or indirectly via a DC $d \in D$. The manufacturer has an access to several third party DCs which are assumed to have unlimited capacity. We assume a shipment cost is incurred for the flows on each outbound arc with a unit variable rate.

3.2 Economic Trends

At the time of making contract decisions, the manufacturer faces uncertainties in market prices, customer demand (both contract and non-contract), customer-contract preferences, raw material prices, and raw material supply availability. These uncertainties are related to the actions of competitors and, in particular, to the state of the economy. In order to take this into account, we assume that the random variables used to model these exogenous factors depend on a finite set Ξ of plausible economic trends over the planning horizon considered. The probability $P(e)$ that economic trend $e \in \Xi$ will prevail over the horizon is estimated subjectively by a panel of industry experts. More specifically, we assume that the probability distribution of the random variables associated to planning period $t \in T$ depends on the prevailing economic trend $e \in \Xi$. The trend is defined by a function of the period index $t \in T$ and it is applied to the value of the original random variables. A typical case would be the consideration of expending, stable and weakening economic trends defined by multiplying a given random variable by a linearly increasing (decreasing) per-period inflation (deflation) factor.

3.3 Customer Contract Policies

As described in Section 1, we examine four potential forms of contracts that the manufacturer may offer to customers: price-only, periodical minimum quantity commitment, periodical commitment with an order band, and periodical stationary commitment. These forms of contracts provide different levels of quantity commitments and flexibilities. For each form of contract, the manufacturer may develop different policies with different contract duration terms and price incentives. Let K^C be the entire set of potential customer contract policies the manufacturer offers. Each contract policy $k \in K^C$ is characterized by a number of distinguishing attributes that influence customer decisions. Without loss of generality, such attributes may include a price discount factor ϕ_k, a fixed contract charge a_k, a quantity flexibility expressed by minimum and maximum quantities lb_k and ub_k, a contract starting period t_k, and a contract duration term N_k (in periods). These attribute values may be determined by the manufacturer's observations of the historical customer ordering behaviours, contract strategies, and pricing experiences. Obviously, the price-only contract provides the greatest quantity flexibility with lb_k being "0" and ub_k being a sufficient large number, while periodical stationary commitment has the least flexibility with $lb_k = ub_k$.

Given the contract commitments and flexibilities, since contract demand may vary randomly, it may be impossible to satisfy the entire contract demand in each period with the finite capacity available. Hence, backlogs are allowed for contract demand. Different backlog penalty costs are

used for different forms of contracts, so that backlog, should it becomes necessary, is more likely to occur for contracts with greater quantity flexibilities (such as price-only contract).

3.4 Customer-Contract Choice Analysis

A manufacturer's decision to offer a contract to a customer does not guarantee that the contract will be signed, but implies that it is feasible and economically advantageous for the manufacturer. Whether or not a customer c will accept a contract k offered, under economic trend e, is modelled using a discrete choice random variable ξ^c_{ke}. In an industrial environment, this choice is affected by many factors such as prices, commitment requirements, customer demand, contract duration terms, product quality, service standards, location, and socio-economic considerations. It is also affected by the competitors' offers available in the market. Let K be the universal contract set offered to some customer population, including all the contract policies offered by the manufacturer, the competitors, as well as the virtual contract ($k = 0$) offered by the spot market, $\left(K \supset K^C\right)$. Each customer c in the customer population has a preference to a subset of the contracts $\mathcal{K}^c \subset K$. According to Ben-Akiva and Lerman (1994) and Vila et al. (2007), the customer's preferences for one contract over the alternative subset of contracts can be modeled based on economic consumer theory, assuming that the customer has the ability to compare all possible contracts, using discrete choice analysis.

In discrete choice analysis, the attractiveness of each alternative contract can be evaluated by a vector of the attribute values, such as $v_k = \left(\phi_k, lb_k, ub_k, N_k, \mathrm{id}(k)\right)$, where id($k$) provides the identity of the manufacturer who is making the offer. Based on random utility theory, the choice preference of customer c for a contract k under economic trend e can be modeled as a linear utility function:

$$U^c_e(k) = \beta_{1e}\phi_k + \beta_{2e} lb_k + \beta_{3e} ub_k + \beta_{4e} N_k + \beta_{5e}\mathrm{id}(k) + \varepsilon_{cek}, \qquad c \in C, e \in \Xi, k \in \mathcal{K}^c$$

where β_{1e}, ..., β_{5e} are parameters to be estimated, and ε_{cek} is an independent Gumbel distributed random disturbance. This random disturbance is introduced to take into account any unexpected influences.

Customer c will likely choose a contract policy $k \in \mathcal{K}^c$ that has the highest utility value. Thus, the probability that customer c chooses a contract k under economic trend e can be expressed by:

$$P^c_e(k) = P\left(U^c_e(k) \geq U^c_e(l), \forall l \in \mathcal{K}^c, l \neq k\right)$$

Note that for a given contract horizon $T_k = \{t_k, ..., t_k + N_k - 1\}$, the manufacturer could only offer a single contract policy $k \in K^C$ to a customer. In order to calculate the probability $P^c_e(k)$, only offer k and offers of the competitors should be considered. Let $\mathcal{K}^c(k) \subset \mathcal{K}^c$ be the set of these

offers. When using a Multinomial Logit discrete choice model, the probability that the contract k would be signed if offered to customer c under economic trend e can be calculated using the expression:

$$P_e^c(k) = \frac{e^{\mu\left(\beta_{1e}\phi_k + \beta_{2e} lb_k + \beta_{3e} ub_k + \beta_{4e} N_k + \beta_{5e}\,\mathrm{id}(k)\right)}}{\sum_{l \in \mathcal{K}^c(k)} e^{\mu\left(\beta_{1e}\phi_l + \beta_{2e} lb_l + \beta_{3e} ub_l + \beta_{4e} N_l + \beta_{5e}\,\mathrm{id}(l)\right)}} \quad, c \in C, e \in \Xi, k \in \mathcal{K}^c(k)$$

where μ is a positive scale parameter.

In order to calculate these probabilities, it is necessary to estimate the parameter values β_{1e}, ..., β_{5e}, $e \in \Xi$. This can be done using revealed preference data (Ben-Akiva and Lerman, 1994) or stated preference data (Louviere *et al.*, 2000). The former is derived from the analysis of each customer's behaviour based on the demand observations available. The later is obtained from a questionnaire with hypothetical offers submitted to a sample of customers. With this data, maximum likelihood estimators are used to obtain the parameter values. This can be implemented, for example, with the BIOGEME software developed by Bierlaire and available on the Web at http://roso.epfl.ch/biogeme. Alternatively, with insufficient customer preference data, subjective preference probabilities $P_e^c(k)$, $e \in \Xi$, may be assigned by the company sales force to each customer c for each contract k.

3.5 *Customer Demand*

When a customer c chooses a contract $k \in K^C$, his demand must comply with the contract duration terms and quantity commitments. We assume that the requirements of customer $c \in C$ for product $i \in I$ during period $t \in T$, under economic trend $e \in \Xi$, is an independent random variable d_{ite}^c with cumulative distribution function $F_{d_{ite}^c}(.)$. Taking into account the contract terms, quantity commitments, and customer choices, the contract demand of customer c under contract k for product i in periods t is defined by:

$$d_{kite}^c = \begin{cases} \min\left(\max\left(lb_k, d_{ite}^c\right), ub_k\right), \text{if } \xi_{ke}^c = 1 \\ 0, \text{otherwise} \end{cases} \qquad \forall i, e, k \in K^C, t \in T_k$$

Note that the contract demand therefore depends on three random variables: the economic trend e, the discrete choice ξ_{ke}^c, and the customer requirements d_{ite}^c. When no contract is signed with customer c for period t, the potential spot demand is assumed to be equal to the customer requirements d_{ite}^c for commodity products $i \in I_{com}$, and to "0" for specialty products $i \in I_{spe}$.

3.6 *Contract and Spot Market Pricing*

In the OSB industry, the manufacturers' contract and spot sales price are influenced by a market reference price, which depends on the economic trend $e \in \Xi$. In order to win customer contracts, manufacturer may use different pricing strategies. For contract pricing, we assume the

manufacturer uses an n-period backward moving average of the market reference price, adjusted by an appropriate contract discount factor ϕ_k. The contract price of product i for customer c under contract k in period t and economic trend e can thus be defined by $p^c_{kite} = \phi_k \sum_{t'=t-n}^{t-1} p^{c,ref}_{it'e} / n$, $\forall c,k,i,t,e$, where $p^{c,ref}_{ite}$ is the market reference price in the customer's market under economic trend e, which is an independent random variable having cumulative distribution function $F_{p^{c,ref}_{ite}}(.)$. For spot sales pricing, we assume the manufacturer uses the market reference price, i.e. $p^c_{ite} = p^{c,ref}_{ite}, \forall c,i,t,e$.

3.7 Supply Contracts and Spot Market Alternatives

At the procurement end, the manufacturer may purchase raw materials from contract suppliers $(s \in CS)$ or on the spot market $(s \in NS)$. At the beginning of each planning horizon, potential contract suppliers offer several supply contracts. Let K^S be the entire set of potential contract policies offered by the suppliers. We assume that suppliers offer only "total minimum quantity commitment" contracts, where each contract policy $k \in K^S$ is characterized by a unique pair of unit purchase cost c^s_{kjt} and total minimum quantity commitment requirement lb^s_k. Alternatively, the manufacturer may purchase raw materials from the spot market at price c^s_{jte}, subject to the market availability KS^s_{te}. The spot market prices and availabilities are assumed to be independent random variables affected by the plausible economic trends, and with cumulative distribution functions $F_{c^s_{jte}}(.)$ and $F_{KS^s_{te}}(.)$.

4. Stochastic Programming Formulation

The superposition of specific realizations of the random variables defined previously gives rise to a set Ω of plausible future scenarios. This is the set of all the scenarios that may occur over the planning horizon under the different plausible economic trends considered. As explained later (in section 6.2), scenarios can be generated using Monte Carlo methods, and a scenario $\omega \in \Omega$ is associated to the following set of specific random variable realizations

$$\left\{\xi^c_k(\omega), d^c_{kit}(\omega), d^c_{it}(\omega), p^c_{kit}(\omega), p^c_{it}(\omega), c^s_{jt}(\omega), KS^s_t(\omega), K_{mt}(\omega), \forall c,s,k,i,j,m,t\right\}$$

We assume that all the contract decisions must be taken at the beginning of the planning horizon, which enables us to model the problem as a two-stage stochastic program with fixed recourse. In the model, the contract decisions (for both demand and supply) are first stage decision variables. In the second stage, future operational decisions and performances are anticipated for given first stage contract decisions, under a given scenario $\omega \in \Omega$. The objective of the model is to find efficient and robust solutions, (1) for the selection of customer demand contracts according to perceived customer choice probabilities, in order to best allocate the manufacturer's capacities; and (2) for the selection of supplier contracts in order to guarantee the satisfaction of the demand.

The model maximizes the manufacturer's expected global profitability while hedging against uncertainty.

4.1 Mathematical Notation

The following notations are required to formulate the model:

Indexes and sets

$m \in M$	Set of manufacturing mills
$c \in C$	Set of customers
$s \in S$	Set of contract (*CS*) and spot (*NS*) raw material suppliers ($S = CS \cup NS$)
$d \in D$	Set of distribution centres (DCs)
$i \in I$	Set of specialty (I_{spe}) and commodity (I_{com}) products ($I = I_{spe} \cup I_{com}$)
$j \in J$	Set of raw materials
$g \in G$	Set of raw material storage technologies
J_g	Set of raw materials requiring storage technology g ($J_g \subset J$)
$r \in \mathcal{R}^{MC}$	Set of outbound arcs from mills to customers ($\mathcal{R}^{MC} = M \times C$)
$r \in \mathcal{R}^{MD}$	Set of outbound arcs from mills to DCs ($\mathcal{R}^{MD} = M \times D$)
$r \in \mathcal{R}^{DC}$	Set of outbound arcs from DCs to customers ($\mathcal{R}^{DC} = D \times C$)
$r \in \mathcal{R}^{O}$	Set of all outbound arcs ($\mathcal{R}^{O} = \mathcal{R}^{MC} \cup \mathcal{R}^{MD} \cup \mathcal{R}^{DC}$)
$k \in K^C$	Set of contract policies the manufacturer offers to customers
$k \in K^S$	Set of contract policies offered by the raw material suppliers
$e \in \Xi$	Set of plausible economic trend over the planning horizon
$t \in T$	Set of planning periods
T_k	Set of planning periods covered by contract k ($T_k \subseteq T$)

Parameters

Sales

$\xi_k^c(\omega)$	Binary choice parameter of customer c for contract policy $k \in K^C$ under scenario ω
a_k	Fixed charge of a demand contract policy $k \in K^C$
a_k^s	Fixed cost of a supply contract with supplier s under contract policy $k \in K^S$
$p_{kit}^c(\omega)$	Sales price of product i for customer c with contract policy k in period t for scenario ω
$p_{it}^c(\omega)$	Spot sales price of product i for customer c in period t for scenario ω
$d_{kit}^c(\omega)$	Contract demand of product i from customer c choosing contract policy k in period t for scenario ω
$d_{it}^c(\omega)$	Spot demand of product i from customer c in period t for scenario ω
π_k	Multiplicative penalty factor for contract $k \in K^C$ demand backlogs

Production

c_{mi}	Unit production cost for product i at mill m
h_{mi}	Inventory holding per unit for product i at mill m
α_{mi}	Capacity consumption coefficient for product i at mill m
$K_{mt}(\omega)$	Production capacity of mill m in period t for scenario ω
u_{mji}	Quantity of raw material j required to produce one unit of product i at mill m
h_{mj}	Unit inventory holding cost of raw material j at mill m
ss_{mj}	Safety stock of raw material j at mill m
KI_m	Finished product storage capacity at mill m (expressed in terms of an upper bound on the inventory level)
KI_{mg}	Raw material storage capacity of technology $g \in G$ at mill m (expressed in terms of an upper bound on the inventory level)

Distribution

c_{ri}	Unit shipping cost for product i on arc r
h_{di}	Inventory holding cost per unit for product i at distribution centre d
tr_{di}	Transhipment cost per unit for product i through distribution centre d

Procurement

c^s_{kjt}	Unit raw material j purchase cost from supplier $s \in CS$ in period t under contract $k \in K^S$
$c^s_{jt}(\omega)$	Unit raw material j spot purchase cost from supplier $s \in NS$ in period t for scenario ω
lb^s_k	Minimum purchase quantity defined by contract policy $k \in K^S$ offered by supplier $s \in CS$
KS^s_t	Supply capacity of contract supplier $s \in CS$ in period t
$KS^s_t(\omega)$	Supply capacity of spot supplier $s \in NS$ in period t for scenario ω
L^s_j	Procurement lead-time of raw material j provided by supplier $s \in S$

Decision variables

First stage variables

Z^c_k	Binary variable equal to "1" if sale contract policy $k \in K^C$ is offered to customer c, and "0" otherwise
Z^s_k	Binary variable equal to "1" if procurement contract $k \in K^S$ is signed with supplier s, and "0" otherwise

Sales recourse variables

$S^c_{it}(\omega)$	Spot sales of product i to customer c in period t for scenario ω

$z_{kit}^{c}(\omega)$ Product i backlog for the demand of customer c under contract k at the end of period t for scenario ω

Production recourse variables

$X_{mit}(\omega)$ Production quantity of product i at mill m in period t for scenario ω

$I_{mit}(\omega)$ Inventory of product i in mill m at the end of period t for scenario ω

$I_{mjt}(\omega)$ Inventory of raw material j at mill m at the end of period t in scenario ω

Distribution recourse variables

$X_{rit}(\omega)$ Quantity of product i shipped on arc r in period t for scenario ω

$I_{dit}(\omega)$ Inventory of product i in distribution centre d at the end of period t for scenario ω

Procurement recourse variables

$X_{kmjt}^{s}(\omega)$ Amount of raw material j purchased by mill m from contract supplier $s \in CS$ under contract $k \in K^{S}$ in period t for scenario ω

$X_{mjt}^{s}(\omega)$ Amount of raw material j purchased by mill m from spot supplier $s \in NS$ in period t for scenario ω

$z_{kj}^{s}(\omega)$ Raw material j procurement underage with respect to the minimum commitment quantity imposed by contract $k \in K^{S}$ with supplier $s \in CS$ for scenario ω

4.2 Scenario Based Stochastic Programming Model

The first stage program is formulated as follows:

$$\max f(\mathbf{Z}) = E_{\Omega}\left[Q(\mathbf{Z},\omega)\right] - \sum_{s \in CS} \sum_{k \in K^{S}} a_{k}^{s} Z_{k}^{s} \tag{4.1}$$

subject to

$$\sum_{k \in K^{C} | t \in T_{k}} Z_{k}^{c} \leq 1 \qquad \forall c,t \tag{4.2}$$

$$\sum_{k \in K^{S}} Z_{k}^{s} \leq 1 \qquad s \in CS \tag{4.3}$$

$$Z_{k}^{c} \in \{0,1\}, \forall c, k \in K^{C}; \qquad Z_{k}^{s} \in \{0,1\}, \forall s, k \in K^{S} \tag{4.4}$$

In the objective function (4.1), $E_{\Omega}[.]$ denotes expected value over all scenarios $\omega \in \Omega$, and $\mathbf{Z}$ the vector of all first stage decision variables Z_{k}^{c} and Z_{k}^{s}. The function $Q(\mathbf{Z},\omega)$, provides the value of the optimal solution of the second stage program for a given $\mathbf{Z}$ and $\omega \in \Omega$. Constraints (4.2) state that the manufacturer cannot have more than one contract with a customer in any period t. Constraints (4.3) state that the manufacturer cannot have more than one contract with a supplier and (4.4) define the domain for the demand and supply contract decision variables.

The objective function of the second stage program is following:

$$Q(\mathbf{Z},\omega) = \max_{\mathbf{Y}(\omega)\geq 0} q(\mathbf{Z},\mathbf{Y}(\omega)) \tag{4.5}$$

$$\begin{aligned} q(\mathbf{Z},\mathbf{Y}(\omega)) = & \left(\sum_{c\in C}\sum_{k\in K^C} \xi_k^c(\omega) a_k Z_k^c\right) + \left(\sum_{c\in C}\sum_{k\in K^C}\sum_{i\in I}\sum_{t\in T_k} p_{kit}^c(\omega) d_{kit}^c(\omega) Z_k^c + \sum_{c\in C}\sum_{i\in I}\sum_{t\in T} p_{it}^c(\omega) S_{it}^c(\omega)\right) \\ & -\left(\sum_{m\in M}\sum_{i\in I}\sum_{t\in T}\left(c_{mi} X_{mit}(\omega) + h_{mi} I_{mit}(\omega)\right)\right) \\ & -\sum_{i\in I}\sum_{t\in T}\left(\sum_{r\in\Re^O} c_{ri} X_{rit}(\omega) + \sum_{r\in\Re^{MD}} tr_{di} X_{rit}(\omega) + \sum_{d\in D} h_{di} I_{dit}(\omega)\right) \\ & -\sum_{m\in M}\sum_{j\in J}\sum_{t\in T}\left(\sum_{s\in CS}\sum_{k\in K^S}\left(c_{kjt}^s X_{kmjt}^s(\omega) + c_{kjT}^s z_{kj}^s(\omega)\right) + \sum_{s\in NS} c_{jt}^s(\omega) X_{mjt}^s(\omega) + h_{mj} I_{mjt}(\omega)\right) \\ & -\left(\sum_{c\in C}\sum_{k\in K^C}\sum_{i\in I}\sum_{t\in T_k} \pi_k p_{kit}^c(\omega) z_{kit}^c(\omega)\right) \end{aligned} \tag{4.6}$$

where $\mathbf{Y}(\omega)$ is the vector of all the second stage decision variables for scenario ω. The function $q(\cdot)$ defines the net profit calculated by summing the revenues from the fixed contract charge, as well as contract and spot sales, as shown in the first two sets of brackets, minus the total cost of production, distribution, procurement, and any penalties as expressed by the third, forth, fifth, and sixth sets of brackets. In the first set of brackets, the fixed contract charge is applied only when the contract is accepted by both parties $\left(\xi_k^c(\omega) = Z_k^c = 1\right)$. In the third set of brackets, the production cost includes the costs of making and inventory holding at the mills. The backlog penalty cost is considered in the last set of brackets. The distribution cost, as shown in the forth set of brackets, consists of the total cost of shipping, transhipment, and inventory holding at the DCs. The procurement cost, as shown in the fifth set of brackets, includes the costs of both contract and non-contract raw material purchases, the inventory holding, as well as the raw material purchase underage $z_{kj}^s(\omega)$, with respect to the contract minimum quantity commitment. The last set of the brackets provides the penalty cost for the backlogs of the contract demand $z_{kit}^c(\omega)$. The recourse variables, $z_{kit}^c(\omega)$ and $z_{kj}^s(\omega)$, ensure the feasibility of the second stage program for all $\mathbf{Z}$.

The second stage program includes the following constraints:

Constraints concerning sales:

$$\sum_{r\in(\Re^{M}\cup\Re^{DC})} X_{rit}(\omega) = \sum_{k\in K^C}\left(Z_k^c d_{kit}^c(\omega) + z_{kit-1}^c(\omega) - z_{kit}^c(\omega)\right) + S_{it}^c(\omega) \qquad \forall c,i,t,\omega \tag{4.7}$$

$$z_{kit}^c(\omega) \leq Z_k^c d_{kit}^c(\omega) \qquad \forall c,i,t,\omega,k\in K^C \tag{4.8}$$

$$S_{it}^c(\omega) \leq (1 - Z_k^c) d_{it}^c(\omega) \qquad \forall c,t,\omega,k\in K^C, i\in I_{com} \tag{4.9}$$

Constraints (4.7) describe the flow balance at a customer node, stating that the shipments to the customer must be equal to the contract sales quantity (if a contract is provided) plus the backlog in the previous period minus the backlog at the end of the current period, or otherwise, the spot sales quantity. Constraints (4.8) provide the bound for the contract demand backlog. Constraints (4.9) state that, when a customer is not served by a contract, spot sales should not exceed the customer's non-contract demand $d_{it}^{c}(\omega)$.

Constraints concerning production and distribution:

$$X_{mit}(\omega)+I_{mit-1}(\omega)-I_{mit}(\omega)=\sum_{r\in(\mathfrak{R}^{MD}\cup\mathfrak{R}^{MC})}X_{rit}(\omega) \qquad \forall m,i,t,\omega \qquad (4.10)$$

$$\sum_{r\in\mathfrak{R}^{MD}}X_{rit}(\omega)+I_{dit-1}(\omega)-I_{dit}(\omega)=\sum_{r\in\mathfrak{R}^{DC}}X_{rit}(\omega) \qquad \forall d,i,t,\omega \qquad (4.11)$$

$$\sum_{i\in I}\alpha_{mi}X_{mit}(\omega)\leq K_{mt}(\omega) \qquad \forall m,t,\omega \qquad (4.12)$$

$$\sum_{i\in I}I_{mit}(\omega)\leq KI_{m} \qquad \forall m,t,\omega \qquad (4.13)$$

Constraints (4.10) and (4.11) are the flow conservation constraints at the mills and the DCs. Constraints (4.12) and (4.13) are capacity constraints for production and inventory, respectively.

Constraints concerning procurement:

$$\sum_{s\in CS}\sum_{k\in K^S}X^{s}_{kmjt-L^s_j}(\omega)+\sum_{s\in NS}X^{s}_{mjt}(\omega)+I_{mjt-1}(\omega)-I_{mjt}(\omega)=\sum_{i\in I}u_{mji}X_{mit}(\omega) \qquad \forall m,j,t=1+L^s_j,\dots,T,\omega \qquad (4.14)$$

$$\sum_{j\in J}\left(\sum_{m\in M}\sum_{t\in T}X^{s}_{kmjt}(\omega)+z^{s}_{kj}(\omega)\right)\geq Z^{s}_{k}lb^{s}_{k} \qquad \forall s\in CS,k\in K^{S},\omega \qquad (4.15)$$

$$\sum_{k\in K^S}\sum_{m\in M}\sum_{j\in J}X^{s}_{kmjt-L^s_j}(\omega)\leq\sum_{k\in K^S}Z^{s}_{k}KS^{s}_{t} \qquad \forall s\in CS,t=1+L^s_j,\dots,T,\omega \qquad (4.16)$$

$$\sum_{m\in M}\sum_{j\in J}X^{s}_{mjt}(\omega)\leq KS^{s}_{t}(\omega) \qquad \forall s\in NS,t,\omega \qquad (4.17)$$

$$\sum_{j\in J_g}I_{mjt}(\omega)\leq KI_{mg} \qquad \forall m,g,t,\omega \qquad (4.18)$$

$$I_{mjt}(\omega)\geq ss_{mj} \qquad \forall m,j,t,\omega \qquad (4.19)$$

Constraints (4.14) are the flow balance constraints for raw material requirements at mills, taking into account the supplier lead times. Constraints (4.15) impose the total minimum quantity commitment the manufacturer must comply with when a supply contract is signed. Constraints (4.16) and (4.17) are capacity constraints for the contract and spot suppliers, respectively. Constraints (4.18) are raw material inventory capacity constraints. Safety stock requirement constraints are given by (4.19).

Valid cuts:

In order to improve the solution time, the following cuts are added to the model:

$$\sum_{m\in M}\left(X_{mit}(\omega)+I_{mit-1}(\omega)-I_{mit}(\omega)\right)+\sum_{d\in D}\left(I_{dit-1}(\omega)-I_{dit}(\omega)\right)=\sum_{c\in C}\left(\sum_{k\in K^c}\left(Z_k^c d_{kit}^c(\omega)+z_{kit-1}^c(\omega)-z_{kit}^c(\omega)\right)+S_{it}^c(\omega)\right)$$

$$\forall i,\ t,\omega \qquad (4.20)$$

These cuts define the aggregate flow balance over the manufacturing sites, DCs and customers. They are valid since they are a linear combination of constraints (4.7), (4.10) and (4.11). Our preliminary tests have shown that the cuts reduce computation time by a factor of about 6.

5. Sample Average Approximation

In most applications, the set Ω includes an infinite number of scenarios, which makes the proposed stochastic programming model impossible to solve. The Sample Average Approximation (SAA) method can however be used to obtain near optimal solutions. This method has been theoretically analysed by several authors (Mak et al., 1999; Sharpiro, 2003) and applied to solve various stochastic supply chain design problems (Santoso et al., 2005; Vila et al., 2007). It involves solving the problem for samples of scenarios randomly selected from the population Ω. For this purpose, B random samples $\Omega_b^N=\{\omega_b^1,...,\omega_b^N\}$, $b=1,...,B$, of N scenarios are generated using Monte Carlo methods. For a sample b, the true problem (4.1) – (4.19), with the expected value function $E_\Omega\left[Q(\mathbf{Z},\omega)\right]$ in (4.1), is replaced by the following SAA program:

$$\max \hat{f}_b^N(\mathbf{Z})=\frac{1}{N}\sum_{n=1}^{N} q\left(\mathbf{Z},\mathbf{Y}(\omega_b^n)\right)-\sum_{s\in CS}\sum_{k\in K^s} a_k^s Z_k^s \qquad (5.1)$$

subject to constraints (4.2) – (4.4) and (4.7) – (4.19).

Note that in this program, the second stage constraints (4.7) – (4.19) are defined over the scenarios $\omega\in\Omega_b^N$ of the sample considered. Program (5.1) is solved for the B samples generated and the best solution found is selected. The SAA program (5.1) is a large mixed integer program (MIP) but, for a moderate sample size N, it can be solved using commercial solvers such as CPLEX. Even if a moderate sample size is used, we expect that the contract decisions made using this approach are considerably more robust than the solutions provided by a deterministic model. Clearly, as the number of scenarios N increases, the quality of the decisions improves. As shown by Shapiro (2003), under mild regularity conditions, the solution of the SAA model converges with probability one to the optimal solution of the true problem, as sample size N increases. Also, using B independent random samples of size N increases the probability of finding the true optimal solution.

An important issue is how to select the best solution among the B solutions found, and how close this solution is to the optimal solution of the true problem. The quality of a candidate solution can be evaluated by estimating a statistical optimality gap and confidence intervals. In the following

paragraphs, we present the SAA solution algorithm developed to solve our model. Similar procedures are found in Santoso et al. (2005) and Vila et al. (2007).

SAA Algorithm:

Step 1. Generate B independent samples of N scenarios Ω_b^N, $b = 1,...B$. For each sample, solve the SAA model (5.1). Let v_b^N and $\hat{\mathbf{Z}}_b^N$ be the corresponding optimal objective value and optimal solution, respectively.

Step 2: Compute the statistical upper bound and variance estimators.

$$\bar{U}_{N,B} = \frac{1}{B}\sum_{b=1}^{B} v_b^N \tag{5.2}$$

It can be shown that $\bar{U}_{N,B} \geq v^*$, where v^* denotes the optimal value of the true problem (Mak et al. 1999). Thus $\bar{U}_{N,B}$ provides a statistical upper bound. Since the B samples generated, and hence $v_1^N,...,v_B^N$, are independent, the variance of $\bar{U}_{N,B}$ is given by:

$$\hat{\sigma}^2_{\bar{U}_{N,B}} = \frac{1}{B(B-1)}\sum_{b=1}^{B}\left(v_b^N - \bar{U}_{N,B}\right)^2 \tag{5.3}$$

Step 3: Compute the statistical lower bound and variance estimators.

For each distinct candidate solution $\hat{\mathbf{Z}}_b^N$ obtained in *Step 1*, estimate the true objective function value $f\left(\hat{\mathbf{Z}}_b^N\right)$ as follows:

$$\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N) = \frac{1}{N_l}\sum_{n=1}^{N_l} Q\left(\hat{\mathbf{Z}}_b^N, \omega^n\right) - \sum_{s\in CS}\sum_{k\in K^s} a_k^s (\hat{Z}_k^s)_b^N \tag{5.4}$$

where $\omega^1,...,\omega^{N_l}$ is a sample of size $N_l \gg N$ generated independently of the samples used to obtain $\hat{\mathbf{Z}}_b^N$ in *Step 1*. Note that $\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)$ is an unbiased estimator of $f\left(\hat{\mathbf{Z}}_b^N\right)$. Since $\hat{\mathbf{Z}}_b^N$ is a feasible solution to the true problem, we have $f\left(\hat{\mathbf{Z}}_b^N\right) \leq v^*$. Thus, $\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)$ provides a lower statistical bound on v^*. Since we have an independent sample, the variance of this estimator is given by:

$$\hat{\sigma}^2_{\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)} = \frac{1}{N_l(N_l - 1)}\sum_{n=1}^{N_l}\left(Q\left(\hat{\mathbf{Z}}_b^N, \omega^n\right) - \sum_{s\in CS}\sum_{k\in K^s} a_k^s (\hat{Z}_k^s)_b^N - \tilde{f}_{N_l}\left(\hat{\mathbf{Z}}_b^N\right)\right)^2 \tag{5.5}$$

Step 4: Calculate the optimality gap and the confidence interval.

Having determined the statistical upper and lower bounds from *Step 2* and *3*, the optimality gap of solution $\hat{\mathbf{Z}}_b^N$ can be estimated by:

$$Gap_{N,B,N_l}\left(\hat{\mathbf{Z}}_b^N\right) = \max\left\{0, \bar{U}_{N,B} - \tilde{f}_{N_l}\left(\hat{\mathbf{Z}}_b^N\right)\right\} \tag{5.6}$$

The estimated variance of the gap is given by:

$$\hat{\sigma}^2_{Gap} = \hat{\sigma}^2_{\bar{U}_{N,B}} + \hat{\sigma}^2_{\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)} \tag{5.7}$$

An approximate $100(1-\alpha)$ percent confidence interval for the optimality gap at $\hat{\mathbf{Z}}_b^N$ is given by:

$$\left[0, Gap_{N,B,N_l}\left(\hat{\mathbf{Z}}_b^N\right) + \frac{t_{\frac{\alpha}{2},B-1}\hat{\sigma}_{\bar{v}_{N,B}}}{\sqrt{B}} + \frac{t_{\frac{\alpha}{2},N_l-1}\hat{\sigma}_{\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)}}{\sqrt{N_l}}\right] \tag{5.8}$$

assuming random variables v_b^N and $Q\left(\hat{\mathbf{Z}}_b^N, \omega^n\right)$ follow a t-distribution with $B-1$ and $N_l - 1$ degrees of freedom, respectively (Mak et al. 1999).

Step 5: Select the solution $\hat{\mathbf{Z}}_b^N, b = 1,..., B$, with the highest estimated true objective function value $\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b^N)$.

Having selected the best contract solution, its quality can be evaluated by examining the gap and confidence interval. If the gap and confidence interval are not acceptable, a larger number of samples B and/or sample size N must be used in order to find better solutions.

6. Application to an OSB Industrial Case

6.1 Case Description

In order to validate the methodology, the two-stage stochastic programming model proposed was applied to the real industrial case context presented in Feng et al. (2008). The numerical tests were based on the field data obtained from a single OSB mill. The mill has a single capacitated production line. Production is carried out in batches using a multi-daylight hot press. The production line produces 11 products, on an MTO basis, and it consumes 8 raw materials supplied by 11 raw material suppliers. The products are sold to 140 customers across 5 different regions in North America. In order to effectively apply the methodology, following a Pareto analysis, 20 customers, accounting for 80% of the sales in the 5 regions, were explicitly considered. The rest of the customers and their demands were aggregated to form the spot market in each of the regions. The shipping unit costs to each of the customers are known, and for the spot markets they were estimated based on the weighted cost to each of the customers within the region. On the raw material procurement side, the lead time varies depending on the suppliers and raw material types, being either 0 or 1 period. For demand contracts, 4 forms of contract were offered to the customers, as described in Section 3.3, with different discount, fixed charges, minimum/maximum quantities, and contract horizons, yielding 28 contract policies. For procurement contracts, we considered 7 supply contract offers from 7 suppliers all with yearly contract duration term. The study was conducted with monthly planning periods and a planning horizon of one year. The scope of the case is outlined in Table 1.

Table 1. The Scope of the OSB Case

Indexes	Sizes
Mills	1
Facilities	1
Distribution centers	2
Products	11
Customers	20
Spot market	5
Raw material suppliers	11
Raw materials	8
Demand contract potential offers	28
Supply contract offers	7
Planning horizon	12 months

In this study, the deterministic parameters were derived from field data as explained in Feng et al. (2008). For the random parameters, probability distributions were estimated respectively, using five years market data for the reference price, and one year's data for customer demand, production capacity, raw material spot price, and raw material spot capacity. The best fit for the market reference price, demand, raw material spot price and raw material spot capacity was a Normal distribution, and the standard deviations were obtained by multiplying the historic means by an estimated coefficient of variation (0.05, 0.20, 0.05, and 0.20, respectively). The best fit for the manufacturing capacity, based on down time analysis, was a Uniform distribution. Three possible economic trends were considered: stable (S), expanding (E), or weakening (W). The corresponding estimated probabilities were $P(S)=70\%$, $P(E)=20\%, P(W)=10\%$ and the trends were defined by linearly increasing (decreasing) annual inflation (deflation) factors $\lambda_{et}=(a_e/T)t+1$ with a_e, being 0%, 10%, and -10%, respectively, for all $e\in\Xi=\{S,E,W\}$, over the planning horizon of T = 12 monthly periods. The distributions for the random variables and corresponding inflation (deflation) factors are shown in Table 2.

Table 2. Random variables, their probability distributions and inflation (deflation) factors

Random Variables	Distributions	inflation/deflation factors
Market reference price $p_{ite}^{c,ref}$	$F_{p_{ite}^{c,ref}}(.)=Normal\left(\mu\left(p_{ite}^{c,ref}\right),\sigma\left(p_{ite}^{c,ref}\right)\right)$	λ_{et}
Demand d_{ite}^{c}	$F_{d_{ite}^{c}}(.)=Normal\left(\mu\left(d_{ite}^{c}\right),\sigma\left(d_{ite}^{c}\right)\right)$	λ_{et}
Raw material spot price c_{jte}^{s}	$F_{c_{jte}^{s}}(.)=Normal\left(\mu\left(c_{jte}^{s}\right),\sigma\left(c_{jte}^{s}\right)\right)$	λ_{et}
Raw material spot capacity KS_{te}^{s}	$F_{KS_{te}^{s}}(.)=Normal\left(\mu\left(KS_{te}^{s}\right),\sigma\left(KS_{te}^{s}\right)\right)$	$\lambda_{et}^{KS}=-\frac{a_e}{T}t+1$
Production capacity K_{mt}	$F_{K_{mt}}(.)=Uniform\left(\theta_1^{K_{mt}},\theta_2^{K_{mt}}\right)$	--

The notations $\mu(.)$ and $\sigma(.)$ are the mean and standard deviation of the Normal variables, θ_1^{K-}, θ_2^{K-} are the lower and upper bounds of the Uniform variables (85% and 98% of the standard production capacity, respectively). Due to the sensitivity of the contract issues on customer-manufacturer-supplier relationships, and a confidentiality agreement, the detailed data is not presented.

The SAA model generator was written using Optimization Programming Language OPL 6.3, with a Microsoft Access database connection to automatically read data inputs and write solution outputs. The MIPs were solved using CPLEX 11.2. The program was run on a Intel Core 2 Duo workstation with CPU 2.00GHz, 4.00GB of RAM, and Windows Vista Home Edition Version 2007.

6.2 Scenario Generation

Plausible scenarios are generated using the following Monte Carlo procedure, which is based on the stochastic processes defined in Section 3. In the procedure, u denotes a uniformly distributed pseudo random number in [0,1]. The procedure starts by selecting an economic trend. It then sequentially generates demands and prices for the customers, capacities and prices for the spot raw material suppliers, and manufacturing capacities. In order to obtain a sample of N scenarios, one simply runs the procedure N times.

Scenario ω Generation Procedure:

Step 1. Select an economic trend e randomly using $P(e), e \in \Xi$

Step 2. For all customers $c \in C$, do:

Generate customer-contract choices

$$\xi_k^c(\omega) = \begin{cases} 1 & \text{if } u \in \left[0, P_e^c(k)\right] \\ 0 & \text{otherwise} \end{cases}, \quad \forall k \in K^C$$

Generate customer requirements and market reference prices

$$d_{it}^c(\omega) = \lambda_{et} F_{d_{it}^c}^{-1}(u), \quad p_{it}^{c,ref}(\omega) = \lambda_{et} F_{p_{it}^{c,ref}}^{-1}(u), \quad \forall i,t \in T$$

Derive contract and spot demands from customer requirements

$$d_{kit}^c(\omega) = \begin{cases} \min\left(\max\left(lb_k, d_{it}^c(\omega)\right), ub_k\right), \text{if } \xi_k^c(\omega) = 1 \\ 0, \text{ otherwise} \end{cases}, \forall i,t \in T_k, k \in K^C$$

$$d_{it}^c(\omega) = \begin{cases} d_{it}^c(\omega) \\ 0 \quad \text{otherwise} \end{cases}, \quad \forall i \in I_{com}, t \in T$$

Derive contract and spot prices from market reference prices

$$p_{kit}^c(\omega) = \frac{\phi_k}{n} \sum_{t'=t-n}^{t-1} p_{it'}^{c,ref}(\omega), \quad \forall i,t \in T_k, k \in K^C$$

$$p_{it}^{c}(\omega)=p_{it}^{c,ref}(\omega)\ ,\ \forall i\in I_{com}, t\in T,$$

Step 3. For all spot suppliers $s\in NS$, and all periods $t\in T$, do:

Generate the spot supplier's raw material capacity and prices

$$KS_t^s(\omega)=\lambda_{st}^{KS}F_{KS_{st}^s}^{-1}(u),\qquad c_{jt}^s(\omega)=\lambda_{st}F_{c_{jst}^s}^{-1}(u)\quad \forall j\in J$$

Step 4. For all mills $m\in M$, and all periods $t\in T$, do:

Generate the manufacturing capacity

$$K_{mt}(\omega)=F_{K_{mt}}^{-1}(u)$$

7. Computational Results

In order to investigate the solvability of the SAA program (5.1), and the quality of the solutions obtained, experiments were initially carried out using 5 samples of scenarios $(B=5)$, each of size N = 1, 5, 10, 15, 20 and 25. Table 3 shows that as the sample size N increases, the SAA program size and the computational times grow significantly. Figure 3 illustrates the time variance in solving the problem for the 5 different samples of the varying sample sizes. Obviously there is a trade-off between the problem size, computational efforts, and solution quality. To obtain good quality solutions while preserving the solvability of the model, we used $B=10$ samples of $N=25$ scenarios in our calculations, yielding 10 candidate solutions.

Table 3. Comparison of Model Complexity with Different Sample Size *N*.

Sample size (*N*)	Continuous variables	Binary variables	Constraints	Time (Sec)
1	6952	177	8570	3
5	40687	245	5338	18
10	84417	264	112189	55
15	140246	294	193378	758
20	204442	354	290038	377
25	270500	385	390118	2223

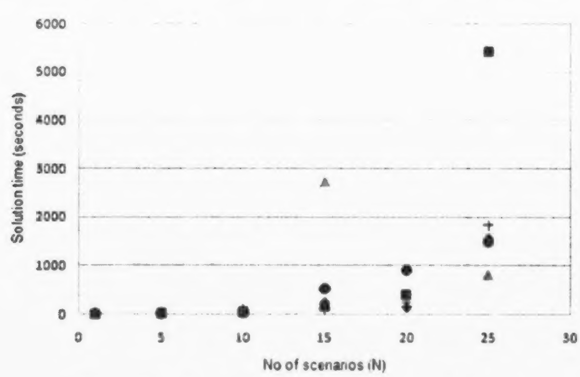

Figure 3. Computation Times Variation.

The statistical validation of the solutions is carried out by evaluating the objective function values with respect to each of the candidate solutions using $N_t = 100, 200, 300$ sampled scenarios. Table 4 presents the statistics for three candidate solutions, denoted $\hat{\mathbf{Z}}_1$, $\hat{\mathbf{Z}}_2$, and $\hat{\mathbf{Z}}_3$. Among the three candidate solutions, the performances are very similar. The objective function values increase and the optimality gaps and confidence intervals become very tight as the sample size N_t increases. This result indicates that the stochastic modeling method can produce robust solutions with good performances in various uncertain environments. The candidate solution $\hat{\mathbf{Z}}_1$ provides slightly superior results among the three solutions. Note that when $N_t = 300$, we observed $\tilde{f}_{N_t}\left(\hat{\mathbf{Z}}_b^N\right) > \bar{U}_{N,B}$ resulting in a negative gap. This negative gap is known to be caused by the separate sampling approach used to estimate the statistical upper and lower bounds, and the relatively smaller values of B and N_t. A similar phenomenon was observed by Mak (1999), where a common random number (CRN) sampling approach was proposed. In the CRN sampling approach, instead of developing a confidence interval of the optimality gap by estimating the upper- and lower- bounds separately using independent sample scenarios, the same set of sample scenarios is used. It was reported that using CRN sampling can eliminate the negative gap with improved confidence interval without significantly increasing the sample sizes.

In order to investigate the necessity of applying stochastic programming in solving contract design, allocation, and selection problems, the problem is also solved using MIP deterministic model. The performances of the contract solutions derived using the two models are then compared. In the deterministic model, the random variables such as demand, market price, raw material supplier price and capacity, as well as the manufacturing capacity, are replaced by mean values under a stable economic environment. The customer-contract choice parameters are generated randomly and independently. Ten replicates of customer-contract choice parameters are generated and the MIP model is solved for each replicate yielding ten candidate solutions. These solutions are also evaluated using $N_t = 100, 200, 300$ sampled scenarios. The performances with respect to the deterministic contract solutions are compared in Figure 4 with those obtained from the stochastic contract solutions. The contract solutions obtained from the stochastic programming model perform significantly better than those obtained from the deterministic model with a 12% performance improvement on average equivalent to $ 7 million dollar increase in profit. The performances from the ten candidate solutions obtained using stochastic programming model are consistent with little variations, while those from the candidate solutions obtained using deterministic model vary considerably, ranging from $53 to $64 million dollars.

Table 4. Stochastic programming model results

Units: Million $				Stochastic					
Point estimate: $\bar{U}_{N,B}$				67.100					
Stdev: $\hat{\sigma}_{\bar{U}_{N,B}}$				0.419					
Candidate solutions:		$\hat{\mathbf{Z}}_1$			$\hat{\mathbf{Z}}_2$			$\hat{\mathbf{Z}}_3$	
Sample size: N_l	**100**	**200**	**300**	**100**	**200**	**300**	**100**	**200**	**300**
Point estimate: $\tilde{f}_{N_l}\left(\hat{\mathbf{Z}}_b\right)$	66.164	67.062	67.371	66.148	67.051	67.368	66.171	67.043	67.352
Standard deviation: $\hat{\sigma}_{\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b)}$	1.075	0.708	0.581	1.090	0.718	0.587	1.070	0.704	0.579
$\text{Gap}_{N,B,N_l}\left(\hat{\mathbf{Z}}_b\right)$	0.936	0.038	0.000	0.952	0.050	0.000	0.930	0.057	0.00 0
Standard deviation: $\hat{\sigma}_{Gap}$	1.494	1.126	1.000	1.509	1.137	1.006	1.489	1.123	0.997
Confidence interval (95%):	[0, 1.447]	[0, 0.436]	[0, 0.365]	[0, 1.467]	[0, 448]	[0, 0.366]	[0, 1.440]	[0, 0.454]	[0, 0.365]
CPU time (minutes):	34	37	41	31	34	38	29	32	36

Table 5. Deterministic model results

Units: Million $				Deterministic					
Candidate solutions:		$\hat{\mathbf{Z}}_{MIP1}$			$\hat{\mathbf{Z}}_{MIP2}$			$\hat{\mathbf{Z}}_{MIP3}$	
Objective function values: $OF_{\hat{Z}_{MIP}}$		68.965			68.685			70.696	
Sample size: N_l	**100**	**200**	**300**	**100**	**200**	**300**	**100**	**200**	**300**
Point estimate: $\tilde{f}_{N_l}\left(\hat{\mathbf{Z}}_b\right)$	63.054	63.956	64.262	60.632	61.683	61.807	60.590	61.440	61.736
Standard deviation: $\hat{\sigma}_{\tilde{f}_{N_l}(\hat{\mathbf{Z}}_b)}$	1.165	0.760	0.608	1.219	0.778	0.643	1.209	0.778	0.629
$\text{Gap}_{N,B,N_l}\left(\hat{\mathbf{Z}}_b\right)$	5.911	5.009	4.703	8.053	7.002	6.879	10.106	9.256	8.960
Standard deviation: $\hat{\sigma}_{Gap}$	1.165	0.760	0.608	1.219	0.778	0.643	1.209	0.778	0.629
Confidence interval (95%):	[0, 6.141]	[0, 5.115]	[0, 4.772]	[0, 8.293]	[0, 7.110]	[0, 6.952]	[0, 10.345]	[0, 9.364]	[0, 9.032]
CPU time (minutes):	7	10	13	8	11	14	8	11	1

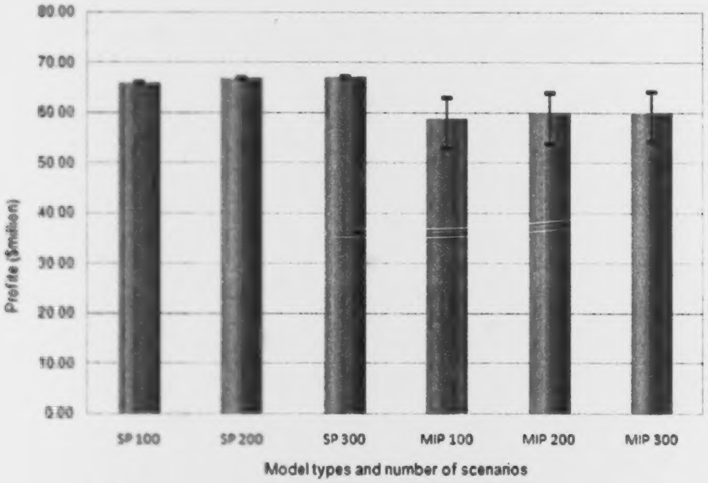

Figure 4. Comparison of objective function values from stochastic and deterministic contract solutions with $N_l = 100, 200, 300$ samples, respectively.

Table 5 present the statistics for three candidate solutions from the deterministic model, denoted $\hat{\mathbf{z}}_{MIP1}$, $\hat{\mathbf{z}}_{MIP2}$, and $\hat{\mathbf{z}}_{MIP3}$. Note that in deterministic case, contract decisions that yield high objective function values do not necessarily perform well in uncertain business situations. As shown in Table 5, particularly candidate solution $\hat{\mathbf{z}}_{MIP3}$, despite an objective function value higher than the upper bound $\bar{U}_{N,B}$ obtained from the stochastic programming model, has a low $\tilde{f}_{N_t}\left(\hat{\mathbf{z}}_{MIP3}\right)$ value and a large gap. Contract decisions are affected by many factors, such as, market price, customer demand and customer-contract choices, among other factors, which are rarely known with certainty. Yet, since in deterministic models, mean parameter values are used and a single customer choice scenario is considered, the decisions made adapt poorly for different plausible scenarios. Thus, decisions provided by deterministic models are less robust, and often inadequate. Stochastic programming is therefore a more appropriate modeling approach for contract decision problems, and the SAA solution approach can be practically applied.

Table 6 presents statistics on the contract decisions provided by the three stochastic and deterministic candidate solutions, respectively. It can be observed that the demand contract decisions vary in terms of contract forms, policies, allocations, and the number of contract customers. This indicates that the decisions are sensitive to the sample scenarios, particularly the customer-contract choices, customer demand, and market prices. This is particularly true for the contract decisions obtained from the deterministic model as shown by the larger variations observed. With the scenarios generated, not all 20 potential high volume customers have been offered a contract. The models have suggested reserving a proportion of the capacity to absorb the contract demand variation and/or serve the spot market. The manufacturer may choose an

alternative contract decisions based on particular contract relationship considerations with full awareness of the financial implications. For the supply contract decisions, the results are rather consistent. Six distinct supply contracts are selected from the 7 contract offers in most of the cases. This indicates that the contract decisions are relatively insensitive for the level of raw material market price and availability changes studied in this case.

Table 6. Candidate solutions.

Candidate solution	No. of contract forms	No of contract policies	Contract demand allocation	No of contract customers	No of supply contract
$\hat{Z}_1$	2	6	63%	16	6
$\hat{Z}_2$	2	7	60%	16	6
$\hat{Z}_3$	3	7	63%	15	6
$\hat{Z}_{MIP1}$	2	9	67%	17	7
$\hat{Z}_{MIP2}$	1	7	67%	18	6
$\hat{Z}_{MIP3}$	4	11	73%	17	6

8. Conclusions and Future Research Opportunities

In this article, we present a two-stage stochastic programming model for coordinated contract design, allocation, and selection decisions from a manufacturer's point of view, in a divergent three-tier manufacturing supply chain, under stochastic economic, market, supply, and system environments. In this capacitated make-to-order manufacturing system, the manufacturer wishes to offer different contracts to satisfy customers' needs, to accept the contract that optimize the resource capacity allocation, and to select the right contracts from the suppliers that guarantee the satisfaction of the contract and non-contract demand at lowest procurement cost. Four forms of contracts are evaluated for the demand contract design including the price-only, periodical minimum quantity commitment, periodical commitment with order band, and periodical stationary commitment contracts, each with different duration terms and price incentives. Stochastic customer-contract choices are incorporated in the scenarios generated in order to provide meaningful solutions for the demand contract decisions. The two-stage stochastic programming model with fixed recourse proposed is solved using the SAA approach. Feasible solutions are obtained in all cases. Computation analysis shows that by using stochastic programming model, more realistic and robust solutions can be obtained, with expected 12% superior financial performances, on average, to those obtained using a MIP deterministic model.

This research has been focused on two-stage stochastic programming to solve a contract decision problem in which all contract decisions are made at the beginning of the planning horizon. In real industrial environments, customer may choose a short term contract, for example, a three month contract, and leave the decisions on future contracts to a later date. A multi-stage stochastic programming approach could thus be investigated to address multiple contract decision points during the planning horizon. Note however that, given the additional complexity introduced by a multi-stage stochastic programming approach, using our model on a rolling horizon basis provides a practical way to reach contract decisions that are near optimal. A comparison of these two approaches would certainly be interesting, despite the fact that the multi-stage models would be much more difficult to solve.

Acknowledgements

The authors would like to gratefully acknowledge the financial support provided by the FOR@C Research Consortium, and would like to thank our research partner, FP Innovations, Forintek division, and our industrial partner for their valuable collaboration and support. We would also like to thank Professor Denis Bolduc for his advises on the application of the BIOGEME software and Mr Philippe Marier for his technical advises.

References

[1] Anupindi, R., 1993, Supply management under uncertainty. PhD thesis, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 199-232.

[2] Anupindi, R., Akella, R., 1997, An inventory model with commitments. Working paper. IN: Tayur, S., Ganeshan, R., Magazine, M. 1999. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, ISBN 0792383443, 9780792383444. pp. 199-232

[3] Anupindi, R., Bassok, Y., 1999, Supply contracts with quantity commitments and stochastic demand. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 199-232

[4] Bassok, Y., Anupindi, R., 1997, Analysis of supply contracts with total minimum commitment. IIE Transactions, 29, 373–381.

[5] Ben-Akiva, M., Lerman, S., 1994. Discrete Choice Analysis: Theory and Application to Travel Demand. MIT, Cambridge, Massachusetts, USA.

[6] Cachon, G.P., 2003. Supply chain coordination with contracts. In: Graves, S.C., de Kok, A.G. (Eds.), Handbooks in Operations Research and Management Science: Supply Chain Management: Design, Coordination and Operation, Vol. 11. North-Holland, Amsterdam, pp. 227-339.

[7] Corbett, C.J., Tang, C.S., 1999, Designing supply contracts: Contract type and information asymmetry. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 271-297.

[8] D'Amours S., Montreuil, B., Moke, J., Desaulniers, G., 2000. Contract-based tactical planning of supply chain, INFORMS Conference, San Antonio, TX, U.S.A.

[9] Easton, F.F., Moodie, D.R., 1999. Pricing and lead time decisions for make-to-order firms with contingent orders. European Journal of Operational Research, 116, 305-318.

[10] Feng, Y., D'Amours, S., and Beauregard, R., 2008a. The value of sales and operations planning in oriented strand board industry with make-to-order manufacturing system: cross functional integration under deterministic demand and spot market recourse. International Journal of Production Economics, 115 (1), 189–209.

[11] Kumar, A., 1992, Supply contracts and manufacturing decisions. PhD thesis, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 199-232.

[12] Lariviere, M.A., 1999, Supply chain contracting and coordination with stochastic demand. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 235-268.

[13] Lariviere, M.A., Porteus, E.L., 2001, Selling to the newsvendor: An analysis of price-only contracts. Manufacturing and Service Operations Management, 3(4), 293-305.

[14] Lian, Z., Deshmukh, A., 2009, Analysis of supply contracts with quantity flexibility. European Journal of Operational Research, 196, 526-533.

[15] Louviere, J., D. Hensher and J. Swait, 2000, Stated Choice Methods, Cambridge University Press.

[16] Mak, W.K., Morton, D.P., Wood, R.K., 1999. Monte Carlo bounding techniques for determining solution quality in stochastic programs. Operations Research Letters, 24, 47-56.

[17] Monahan, J.P., 1984, A quantity discount pricing model to increase vendor profits. Management Science, 30(6), 720-726.

[18] Moinzadeh, K., Nahmias, S., 2000, Adjustment strategies for a fixed delivery contract. Operations Research, 48(3), 408-423.

[19] Munson, C.L., Rosenblatt, M.J., 2001. Coordinating a three-level supply chain with quantity discounts. IIE Transactions 33 (5), 371–384.

[20] Santoso, T., Ahmed, S., Goetschalckx, M, Shapiro, A., 2005, A stochastic programming approach for supplychain network design under uncertainty. European Journal of Operational Research, 167, 96-115.

[21] Scheller-Wolf, A., Tayur, S., 1998, A Markovian Dual-source Production Inventory Model with Order Bands. Working Paper, #1998-E200, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 199-232.

[22] Schneeweiss, C., Zimmer, K., Zimmermann, M., 2004. The design of contracts to coordinate operational interdependencies within the supply chain. International Journal of Production Economics, 92, 43-59.

[23] Shapiro, A., 2003. Monte Carlo sampling methods. IN: Ruszczynski, A., Shapiro, A., (eds), Stochastic Programming, Handbooks in Operations Research and Management Sciences, 10, Elsevier.

[24] Tsay, A.A., 1999, The quantity flexibility contract and supplier-customer incentives. Management Science, 45, 1339–1358.

[25] Tsay, A.A., Lovejoy, W.S., 1999 Quantity flexibility contracts and supply chain performance. Manufacturing and Service Operations Management, 1(2), 89–111.

[26] Tsay, A.A., Nahmias, S., Agrawal, N., 1999, Modeling supply chain contracts: A review. In: Tayur, S., Ganeshan, R., Magazine, M. (Eds.) Quantitative Models for Supply Chain Management. Kluwer Academic Publishers, Boston, pp. 300-336.

[27] van Delft, Ch., Vial, J.-Ph. (2001). Quantitative analysis of multi-periodic supply chain contracts with options via stochastic programming. Working Paper, Department of manangement Studies, University of Geneva, 40 Bd du Pont d'Arve, CH-1211 Geneva 4, Switzerland.

[28] Vila, D., Martel, A., Beauregard, R., 2007. Taking market forces into account in the design of production-distribution networks: A positioning by anticipation approach. Journal of Industrial and Management Optimization, 3(1), 29-50.

[29] Watanapa, B., Techanitisawad, Anulark, 2004. Simultaneous price and due date settings for multiple customer classes. European Journal of Operational Research, doi:10.1016/j.ejor. 2004.02.011

[30] Weng, Z.K., 1995, Channel coordination and quantity discounts. Management Science, 41(9), 1509-1522.

[31] Xu, N., Nozick, L., 2008, Modeling supplier selection and the use of options contracts for global supply chain design. Computers & Operations Research, doi:10.1016/j.cor.2008.12.013

[32] Zou, X., Pokharel, S., Piplani, R., 2008, A two-period supply contract model for a decentralized assembly system. European Journal of Operational Research, 187, 257-274.

CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

An Incremental Tabu Search Heuristic for the Generalized Vehicle Routing Problem with Time Windows

Luigi Moccia
Jean-François Cordeau
Gilbert Laporte

March 2010

CIRRELT-2010-12

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

An Incremental Tabu Search Heuristic for the Generalized Vehicle Routing Problem with Time Windows

Luigi Moccia[1], Jean-François Cordeau[2,3,*], Gilbert Laporte[2,4]

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Richerche, Via P. Bucci 41C, 87036 Rende (CS), Italy

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[3] Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[4] Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. This paper describes an incremental neighbourhood tabu search heuristic for the generalized vehicle routing problem with time windows. The purpose of this work is to offer a general tool that can successfully be applied to a large variety of specific problems. The algorithm builds upon a previously developed tabu search heuristic by replacing its neighbourhood structure. The new neighbourhood is exponential in size, but the proposed evaluation procedure has polynomial complexity. Computational results are presented and demonstrate the effectiveness of the proposed approach.

Keywords. Generalized vehicle routing problems, time windows, tabu search, large neighbourhood search.

Acknowledgements. This work was partly supported by the MUR (Italy) under project LOGIN, and by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 227837-00 and 39682-05. This support is gratefully acknowledged. We thank Stefan Ropke for providing the GVRP instances, and for fruitful discussions.

---

* Corresponding author: Jean-Francois.Cordeau@cirrelt.ca

Dépôt légal – Bibliothèque nationale du Québec,
Bibliothèque nationale du Canada, 2010

Introduction

The purpose of this paper is to describe an incremental tabu search heuristic for the generalized vehicle routing problem with time windows (GVRPTW) defined as follows. Let $G = (V, A)$ be a directed graph, where $V = \{0, 1, ..., n\}$ is the vertex set and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the arc set. Vertex 0 is the depot at which is based a fleet of m identical vehicles, and the vertex set $V \setminus \{0\}$ represents customer sites. The set $V \setminus \{0\}$ is the union of p usually disjoint clusters $V_1, ..., V_p$. Cluster V_h represents the sites of customer h, and the vertices in V_h are characterised by the following customer attributes: a non-negative load q_h, a non-negative service duration d_h, and a time window $[e_h, l_h]$. A travel cost matrix c_{ij} and a travel time matrix t_{ij} are defined on A. The GVRPTW consists in designing at most m routes on G such that: (i) every route starts and ends at the depot; (ii) exactly one site per customer is visited; (iii) the total load and duration of a route r do not exceed Q_r and D_r, respectively; (iv) service at customer h begins within the interval $[e_h, l_h]$; (v) every vehicle leaves the depot and returns to the depot within the interval $[e_0, l_0]$; and (vi) the total routing cost is minimised. The GVRPTW reduces to the classical vehicle routing problem with time windows (VRPTW) when all clusters are singletons, and to the generalized traveling salesman problem with time windows (GTSPTW) when $m = 1$.

This model can be useful for a wide variety of applications, even when the time windows constraints are not considered: see Baldacci et al. (2010) for applications of the GVRP, and Laporte et al. (1996) for applications of the GTSP. Observe that a generalized version of a routing problem arises when customer visits can equivalently take place at more than one site. This gives rise to a location-routing problem because the sites to visit and the vehicle routes must be determined simultaneously (Laporte, 1988).

To the best of our knowledge, there is no literature on the GVRPTW. The closest study, by Bektaş et al. (2009), introduces a branch-and-cut algorithm and an adaptive large neighbourhood search algorithm for the GVRP. In previous articles, Cordeau et al. (2001, 2004) and Cordeau and Laporte (2001) have introduced a unified tabu search (UTS) heuristic which was successfully applied to the VRPTW and several of

its extensions such as the multi-depot VRPTW (MDVRPTW), the periodic VRPTW (PVRPTW), and the site-dependent VRPTW (SDVRPTW). The flexibility of UTS has motivated us to adapt it to generalized vehicle routing problems.

With respect to the existing literature, this paper makes three main contributions. First, it introduces an effective heuristic algorithm for a new problem. Second, it describes how a previous heuristic for vehicle routing problems can be efficiently extended to handle generalized versions of the same problems. A key feature of our heuristic is the use of an incremental procedure to compute successive neighbourhoods. Third, it demonstrates through extensive computational experiments that the proposed general algorithm is competitive with specialised ones.

The remainder of the paper is organised as follows. The proposed algorithm, called incremental tabu search (ITS), is presented in the next section and assessed through extensive computational experiments. Finally, we report some conclusions.

The incremental tabu search algorithm

The ITS heuristic applies the same tabu search mechanisms as UTS but uses a new neighbourhood structure, which can be regarded as a generalisation of the one used in UTS. In the following, we first synthesise the UTS heuristic and its neighbourhood structure, and we then describe the new neighbourhood used in ITS. We will use for the VRPTW the notation defined in the introduction, and we note that $p = n$ for the VRPTW.

The unified tabu search heuristic and its neighbourhood structure

The UTS heuristic moves at each iteration from a solution x to another solution in its neighbourhood $N(x)$. In the following we denote as a solution a set of m routes starting and ending at the depot, and such that each customer is visited by one and only one vehicle. Hence, UTS considers a solution space where time window, duration, and capacity constraints may be violated. These constraints are handled in the objec-

tive function by means of penalty terms equal to the infeasibility value multiplied by a self-adjusting coefficient. If a solution is feasible with respect to any of these constraints, then the corresponding penalty is zero. The penalty coefficients are dynamically adjusted to produce a mix of feasible and infeasible solutions. Thus this relaxation mechanism facilitates the exploration of the search space and is particularly useful for tightly constrained instances. When customer i is removed from route r, its reinsertion in that route is forbidden for the next θ iterations, where θ is a parameter expressing the length of the tabu tenure. However, through an aspiration criterion, a forbidden reinsertion can be performed if this would allow reaching a solution of smaller cost than that of the best known solution containing customer i in route r. A continuous diversification mechanism is also implemented in UTS. The UTS algorithm performs a preset number of iterations η, and returns the best known feasible solution.

In the following we analyse the computational complexity of the UTS neighbourhood structure. The neighbourhood $N(x)$ consists of the solutions that can be obtained by moving each customer from its current route in x to another route. When evaluating the insertion of a customer in a different route all the positions in the new route are considered. Thus a full evaluation of the neighbourhood $N(x)$ computes p customer removals and $p(m-1)$ insertions. If we index a route by $r \in \{1, ..., m\}$, and we express as $C_r(x)$ the set of customers belonging to route r in solution x, then the insertion of a customer in route r requires the assessment of $|C_r(x)| + 1$ new customer sequences. Therefore, the $p(m-1)$ customer route insertions result in the evaluation of $\sum_{r=1}^{m}(|C_r(x)| + 1) \cdot (p - |C_r(x)|)$ customer sequences. Denoting by λ the number of customer sequences to be evaluated because of removal and insertions of customers, we obtain $\lambda = p + \sum_{r=1}^{m}(|C_r(x)| + 1) \cdot (p - |C_r(x)|)$. This leads to the following characterisation of the complexity of fully evaluating $N(x)$.

Proposition 1 *In the worst case, the full evaluation of the neighbourhood $N(x)$ has a computational cost of $O(n^2)$, and this occurs whenever the number of customers per route in the solution x is constant among the routes, hence equal to p/m.*

Proof — We can rewrite λ as $p + \sum_{r=1}^{m}(p|C_r(x)| - |C_r(x)|^2 + p - |C_r(x)|)$, and since

$\sum_{r=1}^{m} |C_r(x)| = p$, we have $\lambda = p^2 + pm - \sum_{r=1}^{m} |C_r(x)|^2$. By applying optimality conditions in the last expression, the term $-\sum_{r=1}^{m} |C_r(x)|^2$, under the constraint $\sum_{r=1}^{m} |C_r(x)| = p$, takes its maximal value $-p^2/m$ whenever $|C_r(x)| = p/m, \forall r \in \{1, ..., m\}$. Recalling that $p = n$ in a standard vehicle routing problem proves the proposition. □

The *full* evaluation of the neighbourhood $N(x)$ is necessary only at the first iteration when the algorithm starts from an initial solution x_0. In fact, when evaluating $N(x_t)$, where x_t is the solution at the iteration $t \geq 1$, we can take advantage of the evaluation of the previous neighbourhood $N(x_{t-1})$. As described above, a solution x_t is obtained by moving a customer from its route in x_{t-1} to a new route in x_t. The impact of removing or inserting customers in the unchanged routes is the same in $N(x_{t-1})$ as it is in $N(x_t)$. Therefore, the evaluation of $N(x_t)$ requires assessing the impact of customer removals and insertions only in the two modified routes. We define this as an *incremental* neighbourhood evaluation. It can be easily verified that the incremental neighbourhood evaluation does not modify the order of magnitude of the worst case computational effort. For this reason, in the following we will refer to the full neighbourhood evaluation of $N(x)$ when discussing the worst case complexity of evaluating the new neighbourhood structure in ITS.

The incremental tabu search neighbourhood structure

In the ITS heuristic the solution space is defined similarly to that of UTS, i.e. we allow violations of time window, duration, and capacity constraints. The new neighbourhood $N'(x)$ consists of all the solutions that can be obtained by moving each customer from its current route to another route, *while considering all the possible site choices for the customers in the modified routes*. We observe that, because of this added feature, $N'(x)$ has an exponential size. Nevertheless, we have devised a neighbourhood evaluation procedure having a polynomial complexity. The idea stems from a well-known result related to the GTSP: given the sequence of clusters in a route, the optimal vertex choice can be determined by shortest path computation in a layered graph; see Renaud and Boctor (1998), and Bontoux et al. (2009) for an application of this result to a memetic

algorithm for the GTSP. Similarly, in the GVRP we can use a layered graph to optimally solve the site selection problem when the customer sequence is fixed in a route. Figure 1 illustrates how this layered graph is generated. We represent the customer sequence in a route by an ordered set $\pi = \{h, k, ..., z\}$, and we introduce a vertex $0'$ corresponding to the departure from the depot, and a vertex $0''$ which denotes the arrival at the depot. The vertex set of the layered graph G_π consists of the vertices $0'$, $0''$, and of the vertices in $\cup_{j\in\pi} V_j$. Vertex $0'$ is linked by arcs only to the vertices corresponding to the sites of the first visited customer, which in our example are the vertices belonging to the set V_h. The cost of each of these arcs is equal to the cost between the depot and the corresponding customer sites, i.e. $c_{0'i} = c_{0i}, \forall i \in V_h$. In turn, the vertices belonging to the set V_h are linked to vertices belonging to the set V_k, etc. These corresponding arcs have costs equal to those of the graph G. Finally, the vertices corresponding to the sites of the last visited customer, i.e. the vertices in V_z, are linked to the arrival vertex $0''$ by arcs with costs $c_{j0''} = c_{j0}, \forall j \in V_z$. The optimal site choice for a given customer

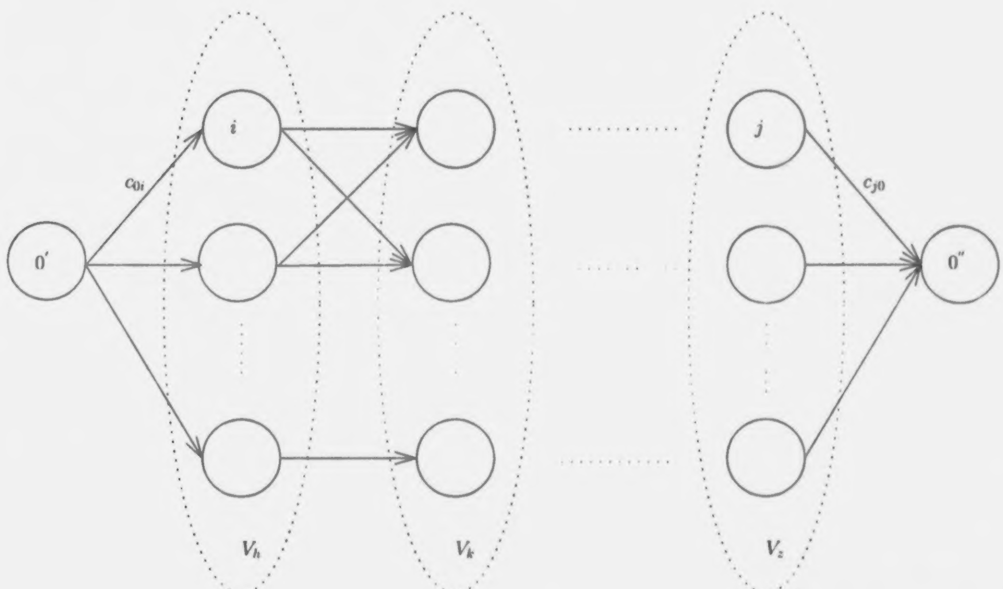

Figure 1: The layered graph G_π for the customer sequence $\pi = (h, k, ..., z)$

sequence π can readily be determined by computing a shortest path from $0'$ to $0''$ in G_π. The cost of this shortest path is the optimal cost of the route visiting the customer

sequence π.

It is well known, see e.g. Ahuja et al. (1993), that a shortest path in an acyclic graph can be computed by the so-called reaching algorithm. This algorithm computes labels from the origin to the destination vertex following their topological order (which in a layered graph does not need to be computed because it can be trivially established). This algorithm has a complexity equal to the number of arcs of the graph. Observe that, analogously to the case described for $N(x)$, the full evaluation of the neighbourhood $N'(x)$ would require the assessment of $O(p^2)$ removals and insertions of customers. Therefore, the reaching algorithm should compute $O(p^2)$ shortest paths. We have to establish worst case conditions for the computation of these shortest paths. Using arguments similar to those of Proposition 1, it can be determined that these worst case conditions occur whenever each route contains p/m customers, and there is a constant number n/p of vertices per cluster. These conditions result in a computational cost per shortest path by the reaching algorithm equal to $O(n^2/(mp))$, which leads to the following result.

Proposition 2 *In the worst case the full evaluation of the neighbourhood $N'(x)$ has a computational cost of $O(p^2 \times n^2/(mp)) = O(n^2p/m)$ when using the reaching algorithm.*

This result indicates that the evaluation of $N'(x)$ should be p/m times more computationally expensive than that of $N(x)$. However, we can significantly speed up computations by taking advantage of the shortest paths computed at previous steps of our algorithm. Indeed, in the exploration of the neighbourhood $N'(x)$, the solutions encountered differ only marginally from the current solution x. The assessment of a solution $y \in N'(x)$ can be regarded as an *incremental optimisation problem* in the sense of Şeref et al. (2009). In the following we show how to considerably reduce the computational effort of this problem. To this end we will employ bi-directional label setting and label recycling as in the work of Hu and Raidl (2008) for the GTSP 2-opt neighbourhood structure.

We define forward labels ϕ as those generated by the reaching algorithm. A label ϕ_i expresses the cost of a shortest path from $0'$ to i in the layered graph. We can

equivalently compute a shortest path from $0'$ to $0''$ by applying the reaching algorithm in reverse mode, i.e. new labels are expanded from $0''$ to $0'$ following their reverse topological order. We define these new labels as backward labels β. Thus, a label β_i indicates the cost of a shortest path from i to $0''$ in the layered graph. By computing both types of labels we obtain the following property:

Property 1 *The cost of the shortest path c^* can be computed for each cluster V as the minimum of the sum of forward and backward labels at each vertex of the cluster,* $c^* = \min_{i \in V}\{\phi_i + \beta_i\}$.

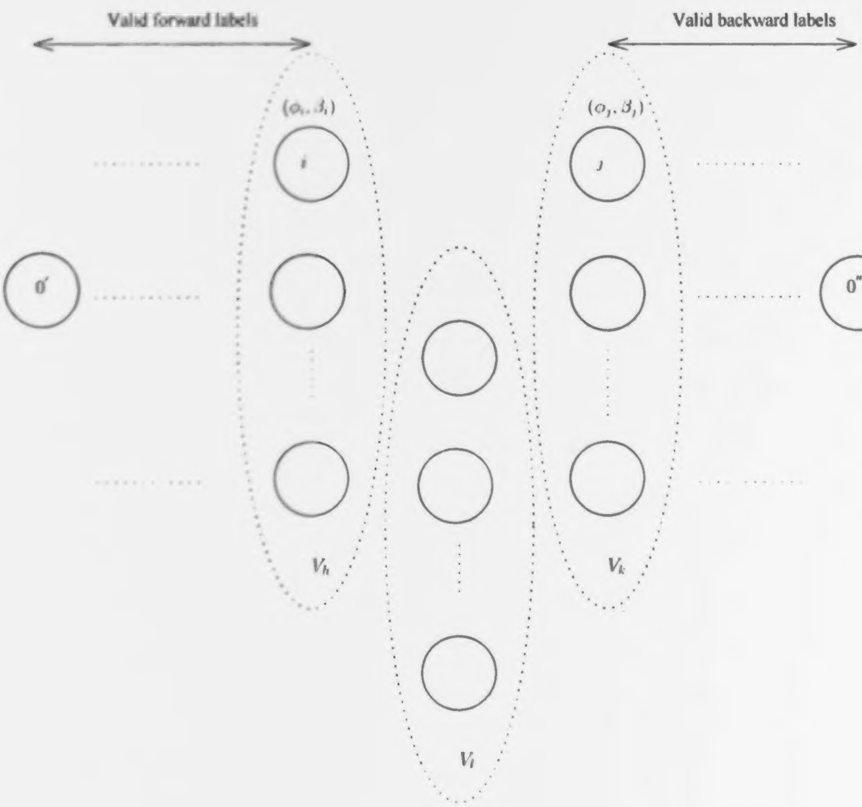

Figure 2: Valid labels when inserting a cluster in a route

As described above, we have to evaluate the insertion of a customer in each position of an existing route. Let $\mu = \{..., h, k, ...\}$ be a customer sequence in a given route. When evaluating the insertion of a customer l after customer h we need the cost of a shortest path from $0'$ to $0''$ in $G_{\mu'}$, where μ' is defined as $\{..., h, l, k, ...\}$, but it is not

necessary to compute a shortest path afresh for each potential insertion. First observe that the forward labels generated by the shortest path computation for the customer sequence μ are still valid for the vertices of the layered graph from vertex $0'$ to the vertices in the set V_h in $G_{\mu'}$. More formally, a label ϕ_i obtained by the shortest path computation in G_μ is valid for the shortest path to be computed in $G_{\mu'}$ if the value of the new forward label for the vertex i in $G_{\mu'}$ is equal to ϕ_i. The label is said to be invalid otherwise. Therefore, the valid labels need not be recomputed, and the reaching algorithm only computes the labels from the vertices in the set V_l to vertex $0''$. Similarly to the previous case, we can recycle the valid backward labels, and when computing the shortest path backward we only compute the backward labels from the vertices in the set V_l to vertex $0'$ (see Figure 2). However, we observe that in order to evaluate the impact of a customer insertion in a given position of a route, we do not have to fully re-evaluate invalid forward and backward labels. Recalling Property 1, we only require *one* cluster in the layered graph with the valid status for both types of labels. In fact, extending the forward labels from the vertices of the set V_h to the vertices of the set V_l and extending the backward labels from the vertices of the set V_k to the vertices of the set V_l suffices to produce a set of valid ϕ and β labels for cluster V_l in $G_{\mu'}$. Hence, the procedure just outlined exactly evaluates the impact of the insertion of customer l between consecutive customers in a route. Its computational burden is proportional to the number of arcs in the layered graph between the vertices of the sets V_h and V_l, and between the vertices of the sets V_l and V_k, i.e. it is equal to $O(|V_h| \cdot |V_l| + |V_l| \cdot |V_k|)$. Under worst case conditions introduced above (customers equally distributed along the routes, and a constant number of vertices per cluster), the insertion procedure with bi-directional label setting and label recycling has a computational cost of $O(2n^2/p^2)$. The procedure needed to evaluate the effect of a customer deletion has a similar computational cost. This result can be derived in way similar to the insertion case, and hence its proof is omitted. We then have the following result:

Proposition 3 *The incremental procedures just introduced bring the worst case computational cost of evaluating $N'(x)$ down to $O(n^2)$.*

Because the complexity of evaluating $N(x)$ in UTS for the standard VRPTW is $O(n^2)$, this result for the GVRPTW indicates that the performance of ITS depends on the number of sites only.

Once a shortest path is computed, we have a choice of vertices for the given customer sequence. The evaluation of the full route is done as in the UTS heuristic, using the forward time slack concept described in Cordeau et al. (2004). We observe that label recycling applies to the shortest path computations only. In fact, a forward label indicates the cost of a shortest path from the origin to the vertex, and is not necessarily equal to the arrival time at the vertex (it can be if $c_{ij} = t_{ij}$, and if the time windows are not binding). The new neighbourhood structure for the GVRPTW can easily be extended to handle periodic, multi-depot, and site-dependent generalized vehicle routing problems, similarly to what is done for the VRPTW in Cordeau et al. (2001), and Cordeau and Laporte (2001).

Computational Results

We now present extensive computational experiments to assess the effectiveness of the ITS algorithm. We have first evaluated ITS on some GVRP instances presented in Bektaş et al. (2009). These instances offer a considerable advantage because tight lower bounds were obtained for them by the branch-and-cut algorithm of Bektaş et al. (2009). Furthermore, in the same article, these instances were solved by an adaptive large neighbourhood search (ALNS) algorithm which is a state-of-the-art metaheuristic. We first provide a comparison of ITS and ALNS on these GVRP instances.

In order to asses the ITS heuristic on instances with time related constraints we have used some MDVRPTW instances. As shown by Baldacci et al. (2010), the MDVRP can be modelled as a GVRP. The solution values obtained by ITS on these MDVRPTW instances were evaluated against results provided by the UTS algorithm. This comparison also provides clues regarding the advantage of a general algorithm versus a specialised one for a given class of problems.

Computational experiments on GVRP instances

The ITS algorithm was coded in ANSI C, and the computational experiments were performed on a desktop computer with an Intel Core Duo 1.83 GHz processor. Our implementation of ITS uses the following procedure to obtain a starting solution x_0: the m routes are initialised by inserting seed customers as distant as possible from each other and from the depot; then, the remaining customers are inserted into routes in a greedy fashion. This procedure was developed to initialise the ITS heuristic in a way similar to that followed in the ALNS algorithm of Bektaş et al. (2009). To evaluate performance of ITS we have used the GVRP instances of Bektaş et al. (2009). In these instances the average number of vertices per cluster is equal to two or three. We refer to Bektaş et al. (2009) for additional details regarding these instances. The ITS heuristic optimally chooses the customer sites given the customer sequence, whereas ALNS does not possess such a feature. We therefore conjecture that ITS should perform better on instances with higher values of n/p, the average number of vertices per cluster in our notation. As we show in the following, the computational experiments reported in Tables 1, 2, 3, and 4, support this conjecture.

These tables are organised as follows. We report in the first column the instance codes as in Bektaş et al. (2009). The second column lists the lower bounds obtained by the branch-and-cut algorithm of these authors. Columns 3 to 5 report the results of their ALNS algorithm, and the last three columns are for the ITS results. We report a solution quality index computed by scaling the heuristic solution value (UB) to the lower bound (LB), where a value of 100.00 corresponds to the lower bound. Thus, an index value of 100.00 indicates an optimal solution. Average solution quality indexes for ALNS and ITS are reported in the last row of the tables.

Computational experiments on small to medium instances with $n/p = 2$ are presented in Table 1. The ALNS algorithm obtains a slightly better average solution quality on these instances. However, the ITS algorithm is slightly better on small to medium instances with $n/p = 3$, see Table 2. As can be seen, ITS consistently reaches optimal or near-optimal solutions. These trends are similar for larger instances, see Tables 3

and 4. On these more difficult instances, the average deviation from the lower bound is just above 5%. We observe that the computational times of the two algorithms are not directly comparable because of different computer processors (Bektaş et al. (2009) used a faster machine). Now, the running time of ITS is proportional to its number η of iterations. Since $\eta = 10^5$ seems necessary to sufficiently explore the solution space, this heuristic seems to be slower than ALNS, even when accounting for the differences in machine speeds.

The comparison presented above was intended to be the toughest possible for ITS, because it is equivalent to testing the UTS algorithm on VRP instances, which is not what UTS was designed for. In fact, GVRP instances do not allow the use of ITS at its full strength because ITS, derived from UTS, works best on routing problems with time windows, and maximal route duration. Still, ITS produces high quality solutions on GVRP instances, and the results obtained on instances with a larger n/p value indicate that this algorithm, although slower than ALNS, is competitive in terms of solution quality with a state-of-the-art metaheuristic expressly designed for the GVRP.

Computational experiments on MDVRPTW instances

As observed by Baldacci et al. (2010), the MDVRP can be modelled as a GVRP on an expanded graph where each customer is represented by a cluster of vertices, one for each depot. This graph contains customer-depot vertices indicated by pairs (h, d), where h is the index of a customer, and d is the index of a depot. An arc connecting two vertices (h, d) and (k, d) related to the same depot d has a cost equal to the cost of the arc between the customers h and k in the original graph. Arcs between vertices related to different depots have a cost equal to a sufficiently large value. The customer-depot vertices are connected to an artificial depot vertex. The cost of an arc connecting the artificial depot to a customer-depot vertex (h, d) is equal to the cost of the arc connecting depot d to customer h in the original graph. This reformulation can also be used to transform a MDVRPTW in a GVRPTW by assigning the time window of each customer h to the corresponding customer-depot vertices (h, d). We observe that this

reformulation exactly transforms a MDVRPTW into a GVRPTW if there are no depot-specific constraints in the MDVRPTW. For example, if there are constraints such as upper bounds on the number of vehicles available at the depots, these constraints do not translate to the standard GVRPTW where an upper bound upon the number of available vehicles can be set for *all* routes. For similar reasons, in the MDVRPTW all depots must have the same time window.

The problem reformulation just described has the drawback of considerably increasing the size of the GVRPTW graph with respect to the size of the original MDVRPTW graph. Observe that the majority of the arcs in the expanded graph, those with an artificial large cost, will never be part of an optimal solution. Thus, testing the ITS algorithm on MDVRPTW instances modelled as GVRPTW poses a remarkable challenge. Still, this reformulation allow us to compare the ITS results with those of UTS. The UTS algorithm is in fact well known for providing high quality solutions on MDVRPTW instances.

Both the ITS and the UTS algorithms were tested on the computer platform used for the previous tests. In this implementation of the ITS algorithm we have used a sweep algorithm to generate an initial solution, which is the default starting procedure of UTS. The instances were derived from those of Cordeau et al. (2001). These new instances differ from those of Cordeau et al. (2001) by having a very large value for the maximum number of routes per depot. This modification was necessary in order to have equivalent instances for both the MDVRPTW and the GVRPTW reformulation. The computational results of UTS and ITS on these MDVRPTW instances are reported in Table 5. Here, we compute the solution quality index by comparing the ITS solution value to the best known solution value. The average solution quality provided by ITS is similar to the one provided by UTS. The computational times of ITS are larger than the ones of UTS, but they remain in the same order of magnitude, in spite of the significant increase in size of the underlying graph. Observe that in these instances the number of depots varies between four and six (third column of Table 5).

These results indicate that a general tool as ITS can be successfully applied to the

solution of specific problems. On the test instances, it consistently yields optimal or near-optimal solutions. Because of the large variety of different location-routing problems that can be modelled as GVRPTW, this is, in our opinion, a noteworthy advantage.

Conclusions

We have presented a new incremental tabu search heuristic, called ITS, for the generalized vehicle routing problems with time windows. The algorithm builds upon a previously developed one by replacing its neighbourhood structure. Although the new neighbourhood is exponential in size, we have presented an evaluation procedure which is polynomial. Computational experiments were performed to assess the effectiveness of the proposed approach. The algorithm provides optimal or near-optimal solutions on instances for which lower bounds are available. In this respect, ITS compares favourably to state-of-the-art metaheuristics.

Acknowledgements

This work was partly supported by the MUR (Italy) under project LOGIN, and by the Natural Sciences and Engineering Council of Canada under grants 227837-00 and 39682-05. This support is gratefully acknowledged. We thank Stefan Ropke for providing the GVRP instances, and for fruitful discussions.

References

Ahuja, R. K., Magnanti, T. L., and Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey.

Baldacci, R., Bartolini, E., and Laporte, G. (2010). Some applications of the generalized vehicle routing problem. *Journal of the Operational Research Society*. Forthcoming.

Bektaş, T., Erdoğan, G., and Ropke, S. (2009). Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. Technical report. http://www.diku.dk/hjemmesider/ansatte/sropke.

Bontoux, B., Artigues, C., and Feillet, D. (2009). A memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem. *Computers & Operations Research*, In Press, Corrected Proof, Available online 18 May 2009.

Cordeau, J.-F. and Laporte, G. (2001). A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR*, 39(3):292–298.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8):928–936.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2004). Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society*, 55(5):542 – 546.

Hu, B. and Raidl, G. (2008). Effective neighborhood structures for the generalized traveling salesman problem. In van Hemert, J. and Cotta, C., editors, *Evolutionary Computation in Combinatorial Optimisation*, volume 4972 of *Lecture Notes in Computer Science*, pages 36–47. Springer, Berlin.

Laporte, G. (1988). Location-routing problems. In Golden, B. L. and Assad, A. A.,

editors, *Vehicle Routing: Methods and Studies*, pages 163–197. North-Holland, Amsterdam.

Laporte, G., Asef-Vaziri, A., and Sriskandarajah, C. (1996). Some applications of the generalized travelling salesman problem. *Journal of the Operational Research Society*, 47(12):1461–1467.

Renaud, J. and Boctor, F. F. (1998). An efficient composite heuristic for the symmetric generalized traveling salesman problem. *European Journal of Operational Research*, 108(3):571 – 584.

Şeref, O., Ahuja, R. K., and Orlin, J. B. (2009). Incremental network optimization: Theory and algorithms. *Operations Research*, 57(3):586–594.

		ALNS			ITS		
Instance	LB	UB	100 × UB/LB	Time (sec.)	UB	100 × UB/LB	Time (sec.)
A-n32-k5-C16-V2	519.0	519	100.00	0.3	519	100.00	9.5
A-n33-k5-C17-V3	451.0	451	100.00	0.3	451	100.00	8.8
A-n33-k6-C17-V3	465.0	465	100.00	0.3	465	100.00	9.0
A-n34-k5-C17-V3	489.0	489	100.00	0.3	489	100.00	9.2
A-n36-k5-C18-V2	505.0	505	100.00	0.3	505	100.00	12.0
A-n37-k5-C19-V3	432.0	432	100.00	[illegible]	432	100.00	[illegible]
A-n37-k6-C19-V3	584.0	584	100.00	0.3	584	100.00	11.1
A-n38-k5-C19-V3	476.0	476	100.00	0.3	476	100.00	11.0
A-n39-k5-C20-V3	557.0	557	100.00	0.4	557	100.00	12.5
A-n39-k6-C20-V3	544.0	544	100.00	0.4	544	100.00	12.6
A-n44-k6-C22-V3	608.0	608	100.00	0.4	608	100.00	15.1
A-n45-k6-C23-V4	613.0	613	100.00	0.4	613	100.00	13.7
A-n45-k7-C23-V4	674.0	674	100.00	0.4	674	100.00	13.7
A-n46-k7-C23-V4	593.0	593	100.00	0.4	593	100.00	13.5
A-n48-k7-C24-V4	667.0	667	100.00	0.4	667	100.00	15.4
A-n53-k7-C27-V4	603.0	603	100.00	0.5	603	100.00	19.4
A-n54-k7-C27-V4	690.0	690	100.00	0.5	690	100.00	18.7
A-n55-k9-C28-V5	699.0	699	100.00	0.5	699	100.00	16.7
A-n60-k9-C30-V5	769.0	769	100.00	0.6	769	100.00	19.3
A-n61-k9-C31-V5	638.0	638	100.00	0.6	638	100.00	20.7
A-n62-k8-C31-V4	740.0	740	100.00	0.6	740	100.00	26.0
A-n63-k10-C32-V5	801.0	801	100.00	0.6	801	100.00	22.7
A-n63-k9-C32-V5	900.3	912	101.30	0.6	912	101.30	21.9
A-n64-k9-C32-V5	763.0	763	100.00	0.6	763	100.00	23.0
A-n65-k9-C33-V5	682.0	682	100.00	0.7	682	100.00	23.3
A-n69-k9-C35-V5	680.0	680	100.00	0.8	680	100.00	26.6
A-n80-k10-C40-V5	957.4	997	104.14	1.0	997	104.14	35.5
B-n31-k5-C16-V3	441.0	441	100.00	0.3	441	100.00	8.2
B-n34-k5-C17-V3	472.0	472	100.00	0.3	472	100.00	9.3
B-n35-k5-C18-V3	626.0	626	100.00	0.3	626	100.00	9.8
B-n38-k6-C19-V3	451.0	451	100.00	0.3	451	100.00	10.7
B-n39-k5-C20-V3	357.0	357	100.00	0.4	357	100.00	11.1
B-n41-k6-C21-V3	481.0	481	100.00	0.4	481	100.00	13.5
B-n43-k6-C22-V3	483.0	483	100.00	0.4	483	100.00	14.8
B-n44-k7-C22-V4	540.0	540	100.00	0.4	540	100.00	12.9
B-n45-k5-C23-V3	497.0	497	100.00	0.5	497	100.00	16.0
B-n45-k6-C23-V4	478.0	478	100.00	0.4	478	100.00	13.8
B-n50-k7-C25-V4	449.0	449	100.00	0.5	449	100.00	16.4
B-n50-k8-C25-V5	916.0	916	100.00	0.5	916	100.00	13.9
B-n51-k7-C26-V4	651.0	651	100.00	0.5	651	100.00	15.1
B-n52-k7-C26-V4	450.0	450	100.00	0.5	450	100.00	16.3
B-n56-k7-C28-V4	486.0	486	100.00	0.6	492	101.23	16.4
B-n57-k7-C29-V4	751.0	751	100.00	0.5	751	100.00	21.5
B-n57-k9-C29-V5	942.0	942	100.00	0.5	942	100.00	18.5
B-n63-k10-C32-V5	816.0	816	100.00	0.6	816	100.00	23.3
B-n64-k9-C32-V5	509.0	509	100.00	0.7	509	100.00	21.8
B-n66-k9-C33-V5	808.0	808	100.00	0.7	808	100.00	24.6
B-n67-k10-C34-V5	673.0	673	100.00	0.7	673	100.00	24.8
B-n68-k9-C34-V5	704.0	704	100.00	0.7	704	100.00	25.7
B-n78-k10-C39-V5	803.0	803	100.00	0.8	804	100.12	34.1
P-n16-k8-C8-V5	239.0	239	100.00	0.2	239	100.00	2.2
P-n19-k2-C10-V2	147.0	147	100.00	0.2	147	100.00	1.7
P-n20-k2-C10-V2	154.0	154	100.00	0.2	154	100.00	1.7
P-n21-k2-C11-V2	160.0	160	100.00	0.2	162	101.25	1.9
P-n22-k2-C11-V2	162.0	162	100.00	0.2	163	100.62	1.9
P-n22-k8-C11-V5	314.0	314	100.00	0.2	314	100.00	3.2
P-n23-k8-C12-V5	312.0	312	100.00	0.2	312	100.00	3.7
P-n40-k5-C20-V3	294.0	294	100.00	0.4	294	100.00	12.6
P-n45-k5-C23-V3	337.0	337	100.00	0.5	337	100.00	15.9
P-n50-k10-C25-V5	410.0	410	100.00	0.5	410	100.00	14.3
P-n50-k7-C25-V4	353.0	353	100.00	0.5	353	100.00	16.1
P-n50-k8-C25-V4	378.4	392	103.60	0.5	421	111.26	16.7
P-n51-k10-C26-V6	427.0	427	100.00	0.5	427	100.00	12.4
P-n55-k10-C28-V5	415.0	415	100.00	0.5	415	100.00	17.3
P-n55-k15-C28-V8	545.3	555	101.78	0.5	565	103.61	13.3
P-n55-k7-C28-V4	361.0	361	100.00	0.6	361	100.00	20.0
P-n55-k8-C28-V4	361.0	361	100.00	0.6	361	100.00	20.0
P-n60-k10-C30-V5	433.0	443	102.30	0.6	443	102.30	20.0
P-n60-k15-C30-V8	553.9	565	102.01	0.6	565	102.01	13.7
P-n65-k10-C33-V5	487.0	487	100.00	0.7	487	100.00	24.1
P-n70-k10-C35-V5	485.0	485	100.00	0.8	485	100.00	27.3
P-n76-k4-C38-V2	383.0	383	100.00	1.0	383	100.00	58.1
P-n76-k5-C38-V3	405.0	405	100.00	0.9	405	100.00	45.7
P-n101-k4-C51-V2	455.0	455	100.00	1.9	455	100.00	119.1
Average			100.20			100.38	

Table 1: Computational results on GVRP small to medium instances with $n/p = 2$. Bold entries correspond to the best solution quality index for each row.

Instance	LB	ALNS UB	ALNS 100 × UB/LB	ALNS Time (sec.)	ITS UB	ITS 100 × UB/LB	ITS Time (sec.)
A-n32-k5-C11-V2	386.0	386	100.00	0.2	386	100.00	4.3
A-n33-k5-C11-V2	315.0	318	100.95	0.2	315	100.00	4.6
A-n33-k6-C11-V2	370.0	370	100.00	0.2	370	100.00	5.7
A-n34-k5-C12-V2	419.0	419	100.00	0.2	419	100.00	6.0
A-n36-k5-C12-V2	396.0	396	100.00	0.2	396	100.00	5.0
A-n37-k5-C13-V2	347.0	347	100.00	0.2	347	100.00	6.1
A-n37-k6-C13-V2	431.0	431	100.00	0.2	431	100.00	7.5
A-n38-k5-C13-V2	367.0	367	100.00	0.2	367	100.00	7.0
A-n39-k5-C13-V2	364.0	364	100.00	0.2	364	100.00	6.8
A-n39-k6-C13-V2	403.0	403	100.00	0.2	403	100.00	7.6
A-n44-k6-C15-V2	503.0	503	100.00	0.3	503	100.00	10.1
A-n45-k6-C15-V3	474.0	474	100.00	0.3	474	100.00	8.5
A-n45-k7-C15-V3	475.0	475	100.00	0.3	475	100.00	8.8
A-n46-k7-C16-V3	462.0	462	100.00	0.3	462	100.00	9.6
A-n48-k7-C16-V3	451.0	451	100.00	0.3	451	100.00	9.8
A-n53-k7-C18-V3	440.0	440	100.00	0.4	440	100.00	12.1
A-n54-k7-C18-V3	482.0	482	100.00	0.4	482	100.00	12.2
A-n55-k9-C19-V3	473.0	473	100.00	0.4	473	100.00	13.7
A-n60-k9-C20-V3	595.0	595	100.00	0.4	595	100.00	16.1
A-n61-k9-C21-V4	473.0	473	100.00	0.4	473	100.00	14.2
A-n62-k8-C21-V3	596.0	596	100.00	0.4	596	100.00	16.9
A-n63-k10-C21-V4	593.0	593	100.00	0.4	593	100.00	14.6
A-n63-k9-C21-V3	625.6	642	102.62	0.4	643	102.78	17.5
A-n64-k9-C22-V3	536.0	536	100.00	0.4	536	100.00	17.2
A-n65-k9-C22-V3	500.0	500	100.00	0.4	500	100.00	18.8
A-n69-k9-C23-V3	520.0	520	100.00	0.4	520	100.00	20.7
A-n80-k10-C27-V4	679.4	710	104.50	0.6	710	104.50	24.9
B-n31-k5-C11-V2	356.0	356	100.00	0.2	356	100.00	4.3
B-n34-k5-C12-V2	369.0	369	100.00	0.2	369	100.00	5.1
B-n35-k5-C12-V2	501.0	501	100.00	0.2	501	100.00	5.7
B-n38-k6-C13-V2	370.0	370	100.00	0.2	370	100.00	7.7
B-n39-k5-C13-V2	280.0	280	100.00	0.2	280	100.00	6.0
B-n41-k6-C14-V2	407.0	407	100.00	0.2	407	100.00	8.5
B-n43-k6-C15-V2	343.0	343	100.00	0.3	343	100.00	9.5
B-n44-k7-C15-V3	395.0	395	100.00	0.3	395	100.00	8.3
B-n45-k5-C15-V2	410.0	422	102.93	0.3	410	100.00	8.4
B-n45-k6-C15-V2	336.0	336	100.00	0.3	336	100.00	10.3
B-n50-k7-C17-V3	393.0	393	100.00	0.3	393	100.00	11.0
B-n50-k8-C17-V3	598.0	598	100.00	0.3	598	100.00	11.0
B-n51-k7-C17-V3	511.0	511	100.00	0.3	511	100.00	10.6
B-n52-k7-C18-V3	359.0	359	100.00	0.3	399	100.00	11.1
B-n56-k7-C19-V3	356.0	356	100.00	0.4	356	100.00	12.6
B-n57-k7-C19-V3	558.0	558	100.00	0.4	558	100.00	13.0
B-n57-k9-C19-V3	681.0	681	100.00	0.4	681	100.00	13.8
B-n63-k10-C21-V3	599.0	599	100.00	0.4	599	100.00	17.2
B-n64-k9-C22-V4	452.0	452	100.00	0.5	452	100.00	15.4
B-n66-k9-C22-V3	609.0	609	100.00	0.4	609	100.00	18.1
B-n67-k10-C23-V4	558.0	558	100.00	0.5	558	100.00	17.0
B-n68-k9-C23-V3	523.0	523	100.00	0.4	523	100.00	20.1
B-n78-k10-C26-V4	606.0	606	100.00	0.5	606	100.00	21.1
P-n16-k8-C6-V4	170.0	170	100.00	0.1	170	100.00	1.5
P-n19-k2-C7-V1	111.0	111	100.00	0.1	111	100.00	0.1
P-n20-k2-C7-V1	117.0	117	100.00	0.1	117	100.00	0.1
P-n21-k2-C7-V1	117.0	117	100.00	0.1	117	100.00	0.1
P-n22-k2-C8-V1	111.0	111	100.00	0.2	111	100.00	0.1
P-n22-k8-C8-V4	249.0	249	100.00	0.2	249	100.00	2.4
P-n23-k8-C8-V3	174.0	174	100.00	0.2	174	100.00	2.9
P-n40-k5-C14-V2	213.0	213	100.00	0.3	213	100.00	7.5
P-n45-k5-C15-V2	238.0	238	100.00	0.3	238	100.00	9.2
P-n50-k10-C17-V4	292.0	292	100.00	0.3	292	100.00	9.2
P-n50-k7-C17-V3	261.0	261	100.00	0.3	261	100.00	10.7
P-n50-k8-C17-V3	262.0	262	100.00	0.3	262	100.00	11.2
P-n51-k10-C17-V4	309.0	309	100.00	0.3	309	100.00	9.6
P-n55-k10-C19-V4	301.0	301	100.00	0.4	301	100.00	11.8
P-n55-k15-C19-V6	378.0	378	100.00	0.4	378	100.00	9.6
P-n55-k7-C19-V3	271.0	271	100.00	0.4	271	100.00	13.3
P-n55-k8-C19-V3	274.0	274	100.00	0.4	274	100.00	13.5
P-n60-k10-C20-V4	325.0	325	100.00	0.4	325	100.00	13.0
P-n60-k15-C20-V5	379.3	382	100.73	0.4	382	100.73	11.6
P-n65-k10-C22-V4	372.0	372	100.00	0.4	372	100.00	15.5
P-n70-k10-C24-V4	385.0	385	100.00	0.5	385	100.00	18.3
P-n76-k4-C26-V2	309.0	320	103.56	0.6	309	100.00	20.3
P-n76-k5-C26-V2	309.0	309	100.00	0.6	309	100.00	27.4
P-n101-k4-C34-V2	370.0	374	101.08	1.0	370	100.00	31.8
Average			*100.22*			100.11	

Table 2: Computational results on GVRP small to medium instances with $n/p = 3$. Bold entries correspond to the best solution quality index for each row.

		ALNS			ITS		
Instance	LB	UB	100 × UB/LB	Time (sec.)	UB	100 × UB/LB	Time (sec.)
M-n101-k10-C51-V5	542.0	542	**100.00**	1.5	542	**100.00**	57.8
M-n121-k7-C61-V4	707.7	719	**101.60**	2.2	720	101.74	98.3
M-n151-k12-C76-V6	629.9	659	**104.62**	3.2	659	**104.62**	113.1
M-n200-k16-C100-V8	744.9	791	**106.19**	5.3	805	108.07	158.7
G-n262-k25-C131-V12	2863.5	3249	**113.46**	6.2	3319	115.91	193.6
Average			**105.18**			*106.07*	

Table 3: Computational results on GVRP large instances with $n/p = 2$. Bold entries correspond to the best solution quality index for each row.

		ALNS			ITS		
Instance	LB	UB	100 × UB/LB	Time (sec.)	UB	100 × UB/LB	Time (sec.)
M-n101-k10-C34-V4	458.0	458	**100.00**	0.9	458	**100.00**	36.8
M-n121-k7-C41-V3	527.0	527	**100.00**	1.2	527	**100.00**	63.3
M-n151-k12-C51-V4	465.6	483	**103.74**	1.9	483	**103.74**	85.5
M-n200-k16-C67-V6	563.1	605	**107.44**	3.0	605	**107.44**	108.4
G-n262-k25-C88-V9	2102.4	2476	117.77	4.9	2463	**117.15**	134.3
Average			*105.79*			**105.67**	

Table 4: Computational results on GVRP large instances with $n/p = 3$. Bold entries correspond to the best solution quality index for each row.

			UTS			ITS		
Instance	Number of customers	Number of depots	UB	100 × UB/Best	Time (sec.)	UB	100 × UB/Best	Time (sec.)
pr01	48	4	1074.1	**100.00**	43	1077.7	100.33	54
pr02	96	4	1738.1	100.07	156	1737.0	**100.00**	245
pr03	144	4	2418.9	100.29	294	2411.9	**100.00**	499
pr04	192	4	2863.0	100.57	432	2846.8	**100.00**	761
pr05	240	4	3047.1	100.64	588	3027.6	**100.00**	917
pr06	288	4	3640.1	**100.00**	767	3698.8	101.61	1328
pr07	72	6	1413.6	**100.00**	93	1414.8	100.08	161
pr08	144	6	2088.7	**100.00**	307	2105.7	100.81	521
pr09	216	6	2727.2	**100.00**	530	2763.8	101.34	1085
pr10	288	6	3515.2	**100.00**	790	3520.4	100.15	1776
pr11	48	4	926.6	**100.00**	62	926.6	**100.00**	77
pr12	96	4	1457.7	101.46	220	1436.8	**100.00**	326
pr13	144	4	2060.7	101.58	375	2028.6	**100.00**	666
pr14	192	4	2251.6	**100.00**	580	2262.3	100.48	905
pr15	240	4	2498.0	**100.00**	674	2503.5	100.22	1186
pr16	288	4	2868.8	**100.00**	993	2936.9	102.38	1580
pr17	72	6	1171.2	**100.00**	127	1171.9	100.06	213
pr18	144	6	1826.6	**100.00**	356	1845.6	101.04	709
pr19	216	6	2294.0	100.08	720	2292.1	**100.00**	1593
pr20	288	6	3112.4	104.29	871	2984.3	**100.00**	1910
Average				100.45			**100.43**	

Table 5: Computational results on MDVRPTW instances. Bold entries correspond to the best solution quality index for each row.